



TUGAS AKHIR - TE 145561

**PERANCANGAN SISTEM *MONITORING NH FUSE* PADA LV
PANEL BERBASIS ARDUINO DENGAN MEDIA *WIFI***

Rizqi Widya Septiarto
NRP 10311500010014

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M. Eng.

PROGRAM STUDI ELEKTRO INDUSTRI
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2018



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

***DESIGN OF NH FUSE MONITORING SYSTEM ON ARDUINO
BASED IN LV PANEL WITH WIFI***

Rizqi Widya Septiarto
NRP 10311500010014

Supervisor
Ir. Josaphat Pramudijanto, M. Eng.

***INDUSTRY ELECTRICAL ENGINEERING
Automation Electrical Engineering Department
Faculty of Vocational
Institut Teknologi Sepuluh Nopember
Surabaya 2018***

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan tugas akhir saya dengan judul “**Perancangan Sistem *Monitoring NH Fuse Pada LV Panel Berbasis Arduino Dengan Media Wifi***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 28 Juli 2018

Rizqi Widya Septiarto
NRP 10311500010014

-----Halaman ini sengaja dikosongkan-----

**PERANCANGAN SISTEM MONITORING NH FUSE PADA LV
PANEL BERBASIS ARDUINO DENGAN MEDIA WIFI**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing

Ir. Josaphat Pramudijanto, M. Eng.

NIP. 1962 10 05 1990 03 1 003

**SURABAYA
JULI, 2018**

-----Halaman ini sengaja dikosongkan-----

PERANCANGAN SISTEM *MONITORING NH FUSE* PADA *LV PANEL* BERBASIS ARDUINO DENGAN MEDIA *WIFI*

Nama : Rizqi Widya Septiarto
NRP : 10311500010014
Pembimbing : Ir. Josaphat Pramudijanto, M. Eng.
NIP : 19621005 199003 1003

ABSTRAK

Kondisi saat ini di PT. PLN (Persero) untuk gangguan *NH fuse* putus belum bisa diketahui secara *real time*. Petugas gangguan mengetahui bahwa *NH fuse* putus setelah adanya laporan dari pelanggan yang daerahnya padam. Petugas terjun langsung ke lapangan, melakukan pengecekan pada *NH fuse*. Hal ini dinilai kurang efisien karena gardu tiang trafo yang harus diinspeksi setiap penyulang jumlahnya ratusan.

Judul tugas akhir ini berupa perancangan sistem *monitoring NH fuse* pada *LV panel* secara *real time*. Cara kerja alat ini adalah mengawasi kerja *NH fuse* pada *LV panel* yang mengambil data arus dan tegangan yang dihasilkan menggunakan sensor. Lalu data dari sensor diproses oleh Arduino dan data yang didapatkan akan dikirim menuju *server* melalui *wifi* dan kinerja *NH fuse* dapat dilihat di komputer *server*.

Pada perancangan alat ini dapat mengukur tegangan dan arus dari *NH fuse*. Hasil dari perancangan ini dapat mengukur tegangan dengan rata-rata nilai kesalahan sekitar 0,07% sampai dengan 0,213% dan pengukuran arus yang mempunyai rata-rata nilai kesalahan antara 3,07% sampai dengan 9,04%. Adapun dilakukan pengujian untuk *monitoring* beban lebih dilakukan dengan mengganti *NH fuse* dengan sekring 1 Ampere. Saat Arus mendekati 1 Ampere, indikator *over current* pada LabVIEW akan menyala. Lalu saat sekring dialiri oleh arus sebesar 1 Ampere yang akan membuat sekring putus. Saat sekring putus membuat indikator status *NH fuse* pada salah satu fasa pada LabVIEW akan mati. Penyesuaian IP *address* pada LabVIEW dapat menunjukkan gardu mana yang saat itu telah di-*monitoring*.

Kata Kunci : *Monitoring, NH fuse, Arduino, Wifi, LabVIEW, Sensor*

-----Halaman ini sengaja dikosongkan-----

DESIGN OF NH FUSE MONITORING SYSTEM ON ARDUINO BASED IN LV PANEL WITH WIFI

Name : Rizqi Widya Septiarto
NRP : 10311500010014
Supervisor : Ir. Josaphat Pramudijanto, M. Eng.
NIP : 19621005 199003 1003

ABSTRACT

Current conditions at PT. PLN (Persero) for breakdown NH fuse can not be known in real time. The disturbance officer learned that NH fuse broke up after a report from an out-of-date customer. Officers plunge straight into the field, checking on NH fuse. This is considered to be less efficient because the transformer pivot beam that must be inspected every hundreds of repeaters.

The title of this final task is the design of NH fuse monitoring system on LV panel in real time. The workings of this tool is to oversee the work of NH fuse on the LV panel which takes the current and voltage data generated using the sensor. Then the data from the sensor is processed by arduino and the data obtained will be sent to the server via wifi and NH fuse performance can be seen on the server computer.

In designing this tool can measure the voltage and current of NH fuse. The results of this design can measure voltages with an average error value of about 0.07% to 0.213% and current measurements having an average error value of between 3.07% to 9.04%. As for testing for monitoring the load more done by replacing the NH fuse with fuse 1 Ampere. Saat Flow approaching 1 Ampere, over current indicator on LabVIEW will light up. Then when the fuse is flowed by a current of 1 Ampere which will make the fuse break. When the fuse breaks make the NH fuse status indicator on any of the phases in LabVIEW will die. IP Address Adjustments on LabVIEW can show which substations are currently being monitored.

Keywords : Monitoring, NH fuse, Arduino, Wifi, LabVIEW, Sensors

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga tugas akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma-3 pada Program Studi Elektro Industri, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis mengucapkan terima kasih kepada orang tua penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Ir. Josaphat Pramudijanto, M. Eng. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya tugas akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian tugas akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada tugas akhir ini. Akhir kata, semoga tugas akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 28 Juli 2018

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL.....	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Sistematika Laporan.....	3
1.6 Relevansi.....	3
BAB II TEORI DASAR	5
2.1 <i>NH Fuse</i> pada <i>LV Panel</i>	5
2.2 Mikrokontroler Arduino Uno	9
2.3 LCD	10
2.4 <i>Ethernet Shield</i>	11
2.5 RTC	12
2.6 Sensor.....	13
2.6.1 Sensor Tegangan	13
2.6.2 Sensor Arus	14
2.7 TP Link TL-WA830RE	15
2.8 Arduino IDE	16
2.9 LabVIEW	16
BAB III PERANCANGAN <i>HARDWARE</i> DAN <i>SOFTWARE</i>	19
3.1 Perancangan <i>Hardware</i>	20
3.1.1 Perancangan Rangkaian <i>Shield</i> Arduino Uno	20
3.1.2 Sensor Tegangan.....	22
3.1.3 Sensor Arus	24

3.1.4 <i>RealTime Clock (RTC) DS1307</i>	24
3.1.5 <i>LCD I2C</i>	25
3.1.6 <i>Ethernet Shield dan Router Wifi</i>	26
3.1.7 <i>Panel Prototype</i>	26
3.2 <i>Perancangan Software</i>	28
3.2.1 <i>Perancangan Program Arduino</i>	28
3.2.3 <i>Flowchart Program</i>	30
BAB IV <i>PENGUJIAN DAN ANALISA DATA</i>	33
4.1 <i>Pengujian Arduino Uno</i>	33
4.2 <i>Pengujian Ethernet Shield</i>	34
4.3 <i>Pengujian Sensor Tegangan</i>	37
4.3 <i>Pengujian Sensor Arus ACS712</i>	51
4.4 <i>Pengujian Software LabVIEW</i>	56
4.5 <i>Pengujian Keseluruhan</i>	57
BAB V <i>PENUTUP</i>	61
5.1 <i>Kesimpulan</i>	61
5.2 <i>Saran</i>	61
DAFTAR PUSTAKA	63
LAMPIRAN A	A-1
A.1. <i>Program Arduino</i>	A-1
A.2. <i>Program LabVIEW</i>	A-8
LAMPIRAN B	B-1
B.1 <i>Datasheet Arduino UNO</i>	B-1
B.2 <i>Datasheet Ethernet Shield</i>	B-6
B.3 <i>Datasheet RTC DS1307</i>	B-8
B.4 <i>Datasheet LCD 20x4</i>	B-14
B.5 <i>Datasheet ACS712</i>	B-16
RIWAYAT HIDUP PENULIS	C-1
UCAPAN TERIMA KASIH	D-1

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	<i>LV Panel</i>	5
Gambar 2.2	Beberapa Tipe <i>NH Fuse</i> Merk <i>GAE</i>	7
Gambar 2.3	Kurva Karakteristik <i>NH Fuse</i> Merk <i>GAE</i>	8
Gambar 2.4	Arduino Uno	10
Gambar 2.5	Konfigurasi Pin LCD I2C	11
Gambar 2.6	Modul <i>Ethernet Shield</i>	12
Gambar 2.7	Modul RTC DS1307	12
Gambar 2.8	Rangkaian RTC DS1307	13
Gambar 2.9	Rangkaian Pembagi Tegangan	14
Gambar 2.10	Modul Sensor Arus ACS712	15
Gambar 2.11	Router TP-link TL WA830RE	15
Gambar 2.12	Arduino IDE	16
Gambar 2.13	LabVIEW 2016	17
Gambar 3.1	Diagram Fungsional Sistem	19
Gambar 3.2	Rangkaian Modul <i>Shield</i> Arduino Uno	21
Gambar 3.3	Hasil Pembuatan <i>Shield</i> Arduino	22
Gambar 3.4	Rangkaian Sensor Tegangan	22
Gambar 3.5	Sensor Tegangan	23
Gambar 3.6	Skema Rangkaian ACS712 30 Ampere	24
Gambar 3.7	Program I2C <i>Scanner</i>	25
Gambar 3.8	Konfigurasi Antara <i>Ethernet shield</i> dan Router	26
Gambar 3.9	<i>Wiring</i> dan Perancangan <i>Prototype</i>	27
Gambar 3.10	Program Deklarasi Pin pada Arduino IDE	28
Gambar 3.11	Desain Tampilan LabVIEW untuk <i>Prototype</i>	29
Gambar 3.12	<i>Block Diagram</i> LabVIEW	30
Gambar 3.13	<i>Flowchart</i> Sistem pada <i>Prototype</i>	31
Gambar 3.14	<i>Flowchart</i> Sistem pada LabVIEW <i>Server</i>	32
Gambar 4.1	Program <i>Analog Read</i> Arduino	33
Gambar 4.2	Hasil <i>Analog Read</i> Arduino	34
Gambar 4.3	Program <i>Web Server</i> Arduino	34
Gambar 4.4	Pembacaan <i>Analog Read</i> yang Dikirim Melalui <i>Wifi</i>	35
Gambar 4.5	Nilai <i>Packet Loss</i> pada Wireshark	36
Gambar 4.6	Nilai <i>Delay</i> pada Wireshark	36
Gambar 4.7	Nilai <i>Throughput</i> pada Wireshark	37

Gambar 4.8	Pengambilan Data Sensor Tegangan	38
Gambar 4.9	Hasil Pembacaan <i>Serial Monitor</i>	38
Gambar 4.10	VariAC 0-250 Volt	39
Gambar 4.11	Data Sensor Tegangan Fasa R	44
Gambar 4.12	Data Sensor Tegangan Fasa S	45
Gambar 4.13	Data Sensor Tegangan Fasa T	45
Gambar 4.14	Pengambilan Data Sensor ACS712	51
Gambar 4.15	Hasil Pembacaan <i>Serial Monitor</i> ACS712	51
Gambar 4.16	Data Linearisasi Sensor Arus Fasa R	53
Gambar 4.17	Data Linearisasi Sensor Arus Fasa S	53
Gambar 4.18	Data Linearisasi Sensor Arus Fasa T	54
Gambar 4.19	Penyalan <i>Prorotype</i> Saat Menguji <i>Software</i>	56
Gambar 4.20	Pengujian <i>Software</i> LabVIEW	57
Gambar 4.21	Pemasangan Sekring pada Salah Satu Fasa	58
Gambar 4.22	Tampilan Indikator Arus dan Status Arus pada <i>NH Fuse</i> Saat Arus Mendekati Batas	58
Gambar 4.23	Tampilan Indikator Status <i>NH Fuse</i> LabVIEW Saat Fasa R Putus	58
Gambar 4.24	Tampilan Lokasi Gardu dan <i>IP Address</i> Saat Program Berjalan	59

DAFTAR TABEL

HALAMAN

Tabel 2.1	Ketentuan Pelebur	6
Tabel 2.2	KHA untuk <i>NH Fuse</i> Sesuai dengan Kapasitas Trafo	6
Tabel 2.3	Tipe <i>NH Fuse</i> Merk <i>GAE</i>	8
Tabel 2.4	Spesifikasi Arduino Uno	9
Tabel 3.1	<i>Mapping Pin Analog/Digital</i> yang digunakan.....	20
Tabel 3.2	Keterangan pada Gambar 3.11	30
Tabel 4.1	Standar Indeks <i>Packet Loss</i>	35
Tabel 4.2	Standar Indeks <i>Delay</i>	36
Tabel 4.3	Standar Indeks <i>Throughput</i>	36
Tabel 4.4	Hasil Pengukuran Sensor Tegangan Fasa R.....	39
Tabel 4.5	Hasil Pengukuran Sensor Tegangan Fasa S	41
Tabel 4.6	Hasil Pengukuran Sensor Tegangan Fasa T	42
Tabel 4.7	Data Perbandingan Tegangan Fasa R	46
Tabel 4.8	Data Perbandingan Tegangan Fasa S	47
Tabel 4.9	Data Perbandingan Tegangan Fasa T.....	49
Tabel 4.10	Data Sensor Arus ACS712 Fasa R.....	52
Tabel 4.11	Data Sensor Arus ACS712 Fasa S	52
Tabel 4.12	Data Sensor Arus ACS712 Fasa T.....	52
Tabel 4.13	Data Perbandingan Sensor Arus Fasa R.....	54
Tabel 4.14	Data Perbandingan Sensor Arus Fasa S	55
Tabel 4.15	Data Perbandingan Sensor Arus Fasa T	55
Tabel 4.16	Data Pengujian Beban Lebih.....	59

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Mengacu pada tugas akhir “Perancangan Sistem *Monitoring Kerja NH Fuse pada LV Panel Menggunakan Mikrokontroler dengan Media Modem GSM*” oleh Elviena Farizka (2013). Menindaklanjuti pentingnya alat ini di dalam distribusi listrik tegangan rendah saya mengembangkan alat ini menggunakan metode yang lebih baru dan efisien yaitu menggunakan arduino dengan media *wifi*.

Distribusi listrik memiliki bidang-bidang yang *include* di dalamnya, antara lain bagian operasional dan proteksi mulai dari *out going* GI (20kV) sampai pelanggan. Proteksi yang ada pada jaringan listrik sangatlah penting untuk melindungi peralatan dan Jaringan Tegangan Rendah adalah *NH fuse pada LV (Low Voltage) panel*. *NH fuse* merupakan pengaman arus lebih yang dipasang untuk melindungi jaringan listrik tegangan rendah 220 Volt. Saat sistem mengalami *over current* (melebihi batas maksimum kerja *fuse*) maka *fuse* akan putus sehingga jaringan listrik putus (*trip*) dan tidak membahayakan peralatan yang lain.

Sampai saat ini, untuk gangguan *NH fuse* (putus) pada JTR (Jaringan Tegangan Rendah) 220 Volt belum dapat terdeteksi secara *real time*. Ketika ada gangguan pada *NH fuse*, petugas gangguan tidak dapat dengan cepat menganalisa lokasi gangguan dan dapat memberi kepastian jawaban atas adanya pengaduan pelanggan yang mengalami padam. Petugas atau *dispatcher* masih menganalisa dan melacak titik lokasi gangguan secara konvensional. Pelacakan ini dengan mencari-cari titik penyebab gangguan. Untuk mendapatkan titik lokasi gangguan dengan cepat, petugas pelayanan teknik yang terjun ke lapangan sangat banyak, dengan biaya transportasi yang juga mahal. Jadi, biaya operasi menangani gangguan masih mahal.

Menindaklanjuti pentingnya alat proteksi ini, diperlukan perhatian khusus untuk menjaga kehandalan sistem kerja dari *NH fuse*. Perlu suatu *monitoring* kerja sehingga ketika *NH fuse* mengalami gangguan (putusnya *fuse*) penanganannya cepat. Dengan menggunakan metode yang lebih baru yaitu menggunakan arduino diharapkan dapat menjalankan sistem *monitoring* dengan baik karena menggunakan teknologi yang lebih baru dan efisien. Dengan dibantunya media pengiriman data menggunakan *wifi* juga akan

mendapatkan koneksi yang lebih stabil dalam menganalisa gangguan antara sistem dan petugas yang berwajib.

1.2 Permasalahan

Permasalahan yang diangkat dari tugas akhir ini adalah karena belum adanya sistem *monitoring* pada *NH fuse* di *LV panel* pada PT. PLN.

1.3 Batasan Masalah

Pada tugas akhir ini, pengolahan data digital yang digunakan adalah arduino uno, dengan memiliki batasan masalah sebagai berikut:

1. *Monitoring NH fuse* 3 fasa, meliputi *telemetering* arus dan tegangan, indikator *overcurrent* dan mendekati batas arus maksimal pada *NH fuse*, data *monitoring* tidak bisa menampilkan dalam bentuk *history*.
2. Parameter yang diukur adalah tegangan 0-250 Volt, dan arus dari 0-2,5 Ampere.
3. Beban yang dicoba pada *telemetering* tidak besar sehingga arus yang mengalir pada *NH fuse* tidak sampai maksimal.

1.4 Tujuan

Tujuan saya menuliskan tugas akhir ini adalah:

1. Merancang sistem *monitoring* yang mampu mengukur tegangan (*out going* trafo distribusi) dan arus *NH fuse* pada *LV panel*, kemudian mengirim hasil pengukuran melalui *wifi* ke komputer *server* dan menampilkan data yang dapat menganalisa status *NH fuse* di indikator pada tampilan LabVIEW.
2. Membuat *prototype* sistem *monitoring* kerja *NH fuse* pada *LV panel* untuk mengetahui status dari *NH fuse tersebut* dan dapat mengukur besarnya tegangan dan arus dalam miniatur 3 fasa dengan pembebanan tidak lebih dari 2,5 Ampere berbasis arduino
3. Mengimplementasikan *monitoring NH fuse* pada *LV Panel* dengan menggunakan *user interface* LabVIEW yang diharapkan dapat membantu petugas PLN dalam menangani gangguan

1.5 Sistematika Laporan

Pembahasan tugas akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, sistematika laporan dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka dari *NH fuse*, *Arduino*, *Ethernet shield*, *LCD 20x4*, *real time clock*, sensor arus ACS 712, sensor tegangan, LabVIEW.

Bab III Perancangan Sistem

Bab ini membahas perencanaan dan pembuatan perangkat keras (*Hardware*) yang meliputi desain alat serta pengimplementasian sensor yang digunakan, pengaturan *Ethernet shield* dan *wifi*, dan pembuatan perangkat lunak (*Software*) yang meliputi program pada *Arduino IDE* untuk menjalankan alat tersebut, serta perancangan *interface* pada LabVIEW.

Bab IV Pengujian dan Analisis

Bab ini memuat tentang pemaparan dan analisis hasil pengujian alat pada keadaan sebenarnya. Seperti pengujian sensor tegangan, dan arus, untuk melihat arus dan tegangan pada *NH fuse*. Selain itu, dilakukan pengujian *wifi*, *RTC*, dan LabVIEW.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Manfaat atau relevansi tugas akhir ini bisa membantu PT. PLN (Persero) dalam pengawasan dan menganalisa gangguan pada *NH fuse* secara *real time* sehingga meningkatkan kehandalan sistem distribusi energi listrik untuk konsumen PT. PLN. Secara tidak langsung juga dapat mengurangi waktu dan biaya yang dihabiskan

petugas PT. PLN dalam penanganan gangguan sehingga konsumen tetap terasa nyaman dan aman.

BAB II

TEORI DASAR

Bab ini membahas mengenai teori - teori yang berkaitan dengan topic penelitian yang dilaksanakan. Teori yang mendukung penyelesaian tugas akhir ini diantaranya ada pemahaman mengenai *NH fuse* pada *LV panel*, arduino, LCD 20x4, *ethernet shield*, sensor tegangan, sensor arus, dan LabView.

2.1 *NH Fuse* pada *LV Panel* [1]

NH Fuse adalah pengaman yang ditempatkan pada PHB (Perangkat Hubung Bagi) tegangan rendah. PHB Tegangan Rendah atau biasa disebut *LV panel* adalah suatu kombinasi dari satu atau lebih peralatan tegangan rendah dengan peralatan kontrol, ukur, pengaman dan pengaturan yang saling berhubungan. Keseluruhannya dirakit lengkap dengan sistem pengawatan dan interkoneksi mekanis serta bagian – bagian penyangganya. *NH fuse* ini sebagai pengaman yang lokasinya di GTT (Gardu Trafo Tiang) seperti terlihat pada Gambar 2.1.



Gambar 2.1 *LV Panel*

Singkatan dari *NH fuse* ini adalah:

N : *Nieder spanung* yang artinya tegangan rendah

H : *Hoch leistung* yang artinya arus besar

Maksudnya *fuse* ini dipergunakan untuk arus yang besar dengan tegangan rendah maksimum 1000 Volt. Arus pengenal yang terpasang di *LV panel* harus dihitung berdasarkan pada kapasitas trafo tegangannya dan Kemampuan Hantaran Arus (KHA). *NH fuse* dipasang untuk mengamankan trafo distribusi dari arus lebih terutama untuk *over load*. Bila terjadi *over load* dalam waktu lama mengenai trafo akan mengakibatkan naiknya temperature isolasi minyak trafo dan bisa mengakibatkan ledakan pada trafo.

Ada standar yang digunakan PT.PLN (Persero) untuk menentukan *NH fuse* yang baik untuk dipasang pada *LV panel*. Standar yang digunakan adalah SPLN 55: 1990, dengan ketentuan pelebur sebagai berikut:

Tabel 2.1 Ketentuan Pelebur

Arus Nominal In (A)	Arus Lebih (A)	Waktu Lebur (Jam)	Catatan
<60	1,3 In	>1	Tidak putus dalam waktu 1 jam
>60	1,3 In	>2	Tidak putus dalam waktu 2 jam
	2 In	<2	Putus dalam waktu maksimum 2 jam

Tabel 2.2 menunjukan KHA untuk *NH fuse* sesuai dengan kapasitas trafo distribusi dengan pembebanan 90% dan jumlah jurusan di sisi tegangan rendahnya.

Tabel 2.2 KHA untuk *NH Fuse* Sesuai dengan Kapasitas Trafo

No	Daya Trafo (KVA)	NH fuse (A).V=400V						
		Jumlah Jurusan						
		1	2	3	4	5	6	7
1	25	35						
2	50	65	35					
3	100	120	60					
4	200	250	120	85	65			
5	315	400	200	135	100			
6	400	510	250	170	125	100	85	
7	630	800	400	275	200	160	135	110
8	1000	1200	600	400	300	240	240	170

Berikut adalah contoh perhitungan KHA *NH fuse* yang digunakan dengan mengetahui kapasitas trafodan jumlah jurusan di sisi tegangan rendah:

Kapasitas trafo : 200 kVA

Ratio tegangan : 20 kV/231-400V

Jumlah jurusan : 4

$$\text{➤ } I_n = \frac{200.000 \text{ VA}}{\sqrt{3} \times 400 \text{ V}} = 288,67 \text{ A}$$

$$\text{➤ Arus tiap jurusan} = 288,67 \text{ A} / 4 = 72,16 \text{ A}$$

$$\text{➤ KHA } NH \text{ fuse yang dipilih} = 72,16 \times 0,9 = 65 \text{ A (beban 90\%)}$$

(Pilih *NH fuse* yang sesuai, sesuaikan KHA dengan ketersediaan *NH fuse*)

Suatu pengaman yang dipilih haruslah memenuhi persyaratan, seperti kecepatan bereaksi, kepekaan operasi (*sensitivity*), selektif dan ekonomis. Maksud dari kecepatan pada *NH fuse* adalah waktu yang dibutuhkan untuk mulai saat adanya gangguan sampai *NH fuse* putus. Kepekaan untuk memberi respon bila ada gangguan. Sedangkan maksud dari selektif adalah kemampuan *NH fuse* untuk menentukan titik dimana gangguan muncul dan memutus rangkaian dengan memutus *fuse* pada *NH fuse*.

Merk *NH fuse* (*NT fuse*) yang biasa digunakan oleh PT. PLN antara lain GAE, Bussman, Kearney, dan Linder. Tiap merk memiliki varian atau tipe yang berbeda-beda. Berikut adalah contoh *NH fuse* merk GAE:



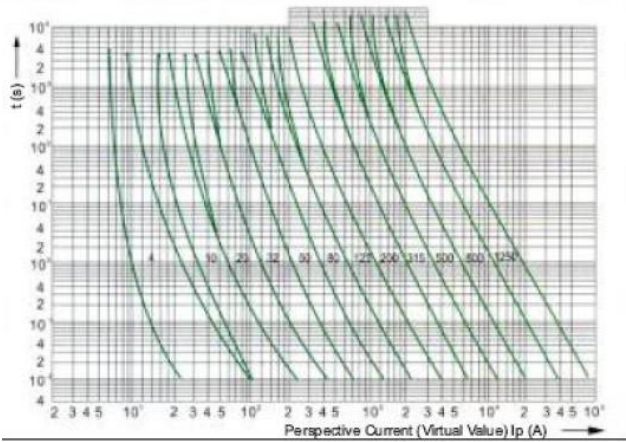
Gambar 2.2 Beberapa Tipe *NH Fuse* Merk GAE

NH fuse merk *GAE* bekerja pada *rating* tegangan sampai 600 Volt, bekerja pada frekuensi 50 Hz. Tabel 2.3 menunjukan *NH fuse* yang dimiliki oleh merk *GAE*.

Tabel 2.3 Tipe *NH Fuse* Merk *GAE*

Model Cross-reference	Rated Voltage (V)	Rated Current (A)	Size	Dimension (mm)					Weight (g)
				A	B	C	D	H	
NT00	500/660	10~160	00	78	50.5	15	30	60	201
NT0	500/660	60~160	0	125	67	15	30	60	267
NT1	500/660	10~160	1	135	67	20	46	58.5	447
NT2	550/660	60~160	2	150	68	20	58.5	68.5	727
NT3	550/660	160~630	3	150	68	36	70	82	975

Mode Is Cross reference (gG) Normal	Rated Voltage (V)	Rated Current (A)	Size	Dimension (mm)								Weight(g)
				A	B	C	D	G	E	F	H	
NT4	500	500~1250	4	200	90	50	97	8	150	16.5	113	2470



Gambar 2.3 Kurva Karakteristik *NH Fuse* Merk *GAE*

Dari Gambar 2.3 ditunjukkan bahwa pelebur *NH fuse* memiliki dua kondisi saat *NH fuse* putus atau melebur, yaitu kondisi minimum dan kondisi maksimum. Maksudnya tiap *NH fuse* memiliki kondisi minimum dan maksimum besar arus yang dapat melebutkan *NH fuse* dalam waktu tertentu.

2.2 Mikrokontroler Arduino Uno [2]

Arduino adalah *board* mikrokontroler *open source* yang terdiri dari *chip* mikrokontroler yang dapat diprogram melalui komputer dengan program arduino IDE. Arduino memiliki port I/O yang dapat membaca sinyal analog dan dapat menulis sinyal digital dari program yang ditanamkan pada *chip* arduino ini.

Tipe Arduino yang digunakan pada pelaksanaan tugas akhir ini adalah Arduino Uno. Arduino Uno adalah mikrokontroler berbasis ATmega328. Arduino Uno memiliki 14 pin digital *input* dan *output* (dimana 6 dapat digunakan sebagai *output* PWM), 6 *input* analog, resonator keramik 16 MHz, koneksi USB, *jack* listrik, *header* ICSP, dan tombol *reset*. Uno dibangun berdasarkan apa yang diperlukan untuk mendukung mikrokontroler, sumber daya bisa menggunakan power USB (jika terhubung ke komputer dengan kabel USB) dan juga dengan adaptor atau baterai. Pada Tabel 2.4 merupakan spesifikasi dari Arduino Uno. Gambar 2.4 adalah bentuk dari Arduino Uno.

Tabel 2.4 Spesifikasi Arduino Uno

Mikrokontroler	ATmega328
Operasi tegangan	5Volt
Input tegangan	disarankan 7-11Volt
Input tegangan batas	6-20Volt
Pin I/O digital	14 (6 bisa untuk PWM)
Pin Analog	6
Arus DC tiap pin I/O	50mA
Arus DC ketika 3.3V	50mA
Memori flash	32 KB (ATmega328) dan 0,5 KB digunakan oleh <i>bootloader</i>
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Kecepatan clock	16 MHz

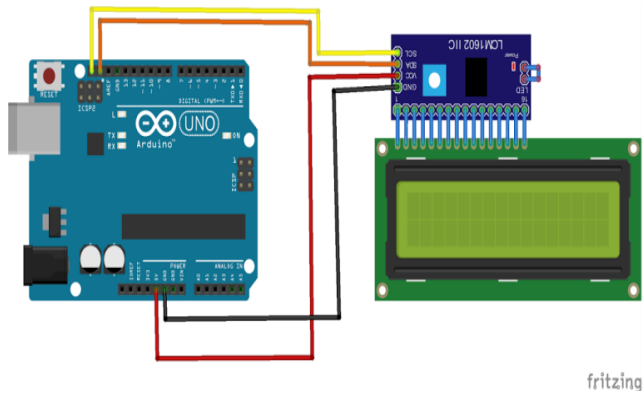


Gambar 2.4 Arduino Uno

2.3 LCD [3]

LCD (*Liquid Crystal Display*) adalah komponen elektronika yang berfungsi untuk menampilkan suatu output berupa nilai atau teks pada mikrokontroler. Sumber cahaya pada LCD adalah lampu neon berwarna putih di bagian belakang susunan cairan kristal cair. Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan ini yang akan membentuk suatu tampilan. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya hanya beberapa warna diteruskan sedangkan warna lainnya disaring.

Pada tugas akhir ini menggunakan LCD 20x4 dan menggunakan modul I2C LCD. Yang dimaksud modul I2C adalah modul LCD yang dikendalikan secara serial sinkron dengan *protocol* I2C/IIC (*Inter Integrated Circuit*) atau TWI (*Two Wire Interface*). I2C hanya membutuhkan 4 port yaitu port SCL (*Serial Clock*), SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya, VCC, dan *ground*. Konfigurasi pin antara LCD I2C dan Arduino Uno dapat dilihat pada Gambar 2.5.



Gambar 2.5 Konfigurasi Pin LCD I2C

2.4 *Ethernet Shield* [4]

Ethernet shield adalah perangkat elektronika untuk Arduino yang digunakan untuk mengkoneksikan Arduino dengan internet. Arduino *ethernet shield* dibuat berdasarkan pada Wiznet W5100 *ethernet chip*. Wiznet W5100 menyediakan IP untuk TCP dan UDP, yang mendukung hingga 4 *socket* secara simultan. Untuk menggunakannya dibutuhkan *library ethernet* dan SPI. Dan *ethernet shield* ini menggunakan kabel RJ-45 untuk mengkoneksikanya ke Internet.

Ethernet shield bekerja dengan cara memberikan layanan IP pada Arduino dan pc agar dapat terhubung ke internet. Cara menggunakan cukup mudah yaitu hanya dengan menghubungkan Arduino *ethernet shield* dengan Arduino lalu akan disambungkan ke jaringan internet. Cukup dengan memasang modul ini ke *board* Arduino, lalu menghubungkannya ke jaringan internet dengan kabel RJ-45, maka Arduino akan terkoneksi langsung ke internet. Dan untuk menggunakannya, tentu saja kita harus menyetting IP pada modul dan pc internet agar dapat terhubung satu sama lain.

Pada *ethernet shield* juga terdapat modul *micro SD card* yang dapat digunakan sebagai sistem penyimpanan data pada Arduino. Modul *micro SD card* ini dapat menyimpan nilai sensor secara *real time* yang disebut juga dengan *data logger*. *Ethernet shield* dapat dilihat pada Gambar 2.6.

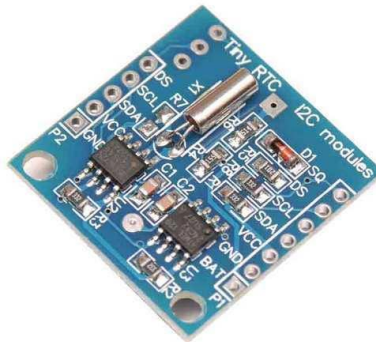


Gambar 2.6 Modul *Ethernet Shield*

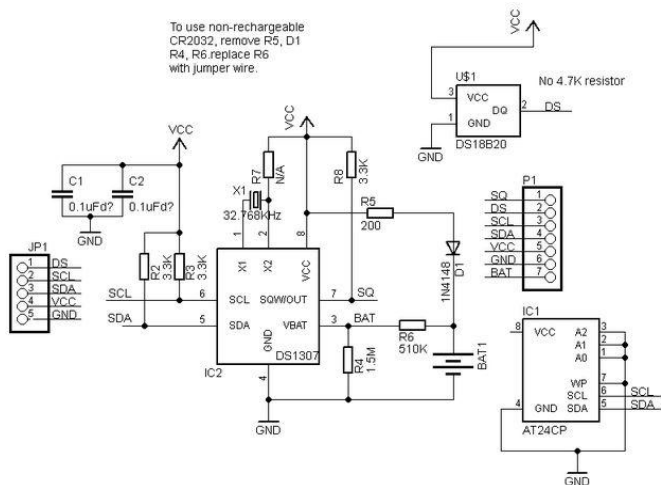
2.5 RTC [2]

RTC (*Real time clock*) adalah jam elektronik berupa *chip* yang dapat menghitung waktu (mulai detik hingga tahun) dengan akurat dan menjaga/menyimpan data waktu tersebut secara *real time*. Karena jam tersebut bekerja *real time*, maka setelah proses hitung waktu dilakukan *output* datanya langsung disimpan atau dikirim ke *device* lain melalui sistem antarmuka.

RTC yang digunakan pada tugas akhir ini adalah RTC DS1307. RTC ini memiliki *supply* alternatif sehingga dapat terus menjaga waktu saat sumber daya dari Arduino tidak ada. *Supply* alternatif ini berasal dari baterai *lithium*. Gambar dari Gambar 2.7 adalah modul RTC DS1307 yang digunakan. Gambar 2.8 adalah rangkaian dari RTC DS1307.



Gambar 2.7 Modul RTC DS1307



Gambar 2.8 Rangkaian RTC DS1307

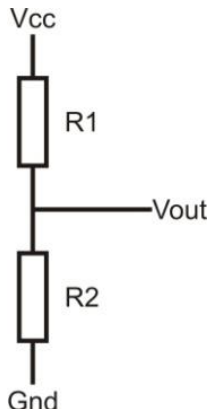
2.6 Sensor

Sensor adalah komponen elektronika yang digunakan untuk mengukur ataupun mendeteksi suatu perubahan fisik atau kimia. Variabel keluaran yang dihasilkan sensor yang dirubah menjadi besaran listrik disebut transduser.

Pada tugas akhir ini, ada dua sensor yang digunakan yaitu sensor tegangan dan sensor arus.

2.6.1 Sensor Tegangan [1]

Sinyal tegangan yang akan diukur dengan menurunkan tegangan input menggunakan *transformator step down* dan rangkaian pembagi tegangan dimana resistor disusun secara seri. Prinsip dari rangkaian pembagi tegangan sesuai dengan hukum *kirchoff* tegangan yang menyatakan bahwa “Tegangan dalam rangkaian tertutup sama dengan jumlah semua tegangan di seluruh rangkaian”. Dari Gambar 2.8, R1 dan R2 dipasang secara seri, di mana tegangan keluaran (V_{out}) adalah tegangan R2. Gambar rangkaian pembagi tegangan ditunjukkan pada Gambar 2.9



Gambar 2.9 Rangkaian Pembagi Tegangan

Berdasarkan Gambar 2.9 dapat diperoleh persamaan sebagai berikut:

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \dots\dots\dots(2.1)$$

Dimana R2 dan R1 merupakan kombinasi resistor yang membagi tegangan input (V_{in}) dan V_{out} merupakan tegangan keluar pada R2. [6]

2.6.2 Sensor Arus [5]

Suatu rangkaian elektronik terdapat tegangan, arus dan hambatan yang saling berhubungan. Untuk itulah pada tugas akhir kali ini sensor arus berperan penting sekali untuk berjalannya sistem *prototype* ini. Sensor arus adalah perangkat elektronika yang dapat mengukur arus dari suatu rangkaian listrik dengan dipasangkan seri dengan rangkaian yang akan diukur.

Sensor arus yang digunakan pada tugas akhir ini adalah sensor arus ACS712. Sensor ini dapat mengukur arus hingga 30 Ampere. Sensor arus ACS712 membutuhkan 3 pin pada Arduino yaitu VCC, pin *input* analog, dan *ground*. Gambar 2.10 adalah gambar dari ACS712.



Gambar 2.10 Modul Sensor Arus ACS712

2.7 TP Link TL-WA830RE [7]

Untuk memenuhi komunikasi melalui *wifi* antara alat dan *server* maka dibutuhkan *router* untuk mengirim data antara alat dan *server*. *Router* disambungkan ke *Ethernet shield* menggunakan kabel RJ-45 untuk mengirim data yang telah diproses oleh Arduino. *Router* yang digunakan adalah router dari TP Link yaitu TL-WA830RE. *Router* ini memiliki kecepatan nirkabel hingga 300Mbps, dengan kecepatan secepat itu diharapkan pengiriman data ke *server* juga bisa lebih cepat dan akurat. TP Link TL-WA830RE dapat dilihat pada Gambar 2.11



Gambar 2.11 Router TP-link TL WA830RE

2.8 Arduino IDE [2]

IDE merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Disebut sebagai lingkungan karena melalui *software* inilah Arduino dilakukan pemrograman untuk melakukan fungsi- fungsi yang dibenamkan melalui sintaks pemrograman. Arduino menggunakan bahasa pemrograman sendiri yang menyerupai bahasa C. bahasa pemrograman Arduino (*sketch*) sudah dilakukan perubahan untuk memudahkan pemula dalam melakukan pemrograman dari bahasa aslinya. Sebelum dijual ke pasaran IC mikrokontroler Arduino telah ditanamkan suatu program bernama *botloader* yang berfungsi sebagai penengah antara *compiler* Arduino dengan mikrokontroler. Arduino IDE dibuat dari bahasa pemrograman JAVA.

Arduino IDE juga dilengkapi dengan *library* C/C++ yang biasa disebut *Wiring* yang membuat operasi *input* dan *output* menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino. Gambar 2.12 adalah tampilan *start up* dari Arduino IDE.



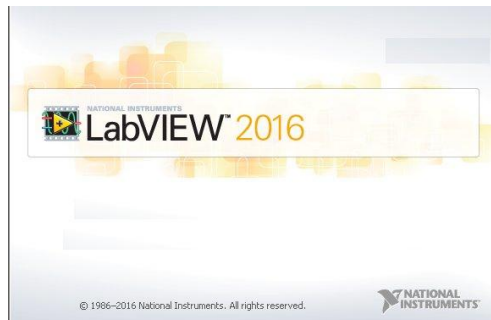
Gambar 2.12 Arduino IDE

2.9 LabVIEW [4]

LabVIEW adalah *software* pemrograman yang diproduksi oleh *National Instruments* dengan konsep yang berbeda. Seperti bahasa pemrograman lainnya yaitu C++, matlab atau *Visual Basic*, LabVIEW juga mempunyai fungsi dan peranan yang sama,

perbedaannya bahwa LabVIEW menggunakan bahasa pemrograman berbasis grafis atau *block diagram* sementara bahasa pemrograman lainnya menggunakan basis *text*.

Program LabVIEW dikenal dengan sebutan VI atau *Virtual Instruments* karena penampilan dan operasinya dapat meniru sebuah *instrument*. Pada LabVIEW, user pertama-tama membuat *user interface* atau *front panel* dengan menggunakan kontrol dan indikator, yang dimaksud dengan kontrol adalah *knobs*, *push button*, *dials* dan peralatan input lainnya sedangkan yang dimaksud dengan indikator adalah *graphs*, LEDs, dan peralatan *display* lainnya. Setelah menyusun *user interface*, lalu *user* menyusun *block diagram* yang berisi kode-kode Vis untuk mengontrol *front panel*. Gambar 2.13 adalah tampilan *start up* dari LabVIEW 2016.



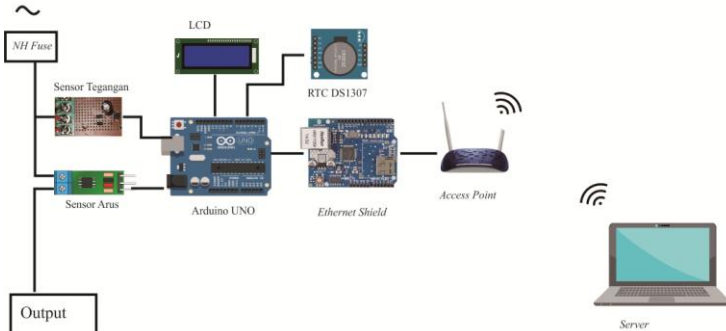
Gambar 2.13 LabVIEW 2016

-----Halaman ini sengaja dikosongkan-----

BAB III

PERANCANGAN *HARDWARE* DAN *SOFTWARE*

Pada perancangan dan pembuatan alat ini terdiri dari pembuatan *hardware* dan *software* yang akan digunakan di rancang bangun *prototype* ini. Bab ini membahas tentang tahapan yang dilakukan terhadap perancangan dan pembuatan *prototype* Tugas Akhir yang berjudul Perancangan Sistem *Monitoring NH Fuse* Pada *LV Panel* Berbasis *Arduino* Dengan Media *Wifi*. Pada pembuatan *hardware* ini meliputi perangkaian antara beberapa komponen penting yang diperlukan, seperti *Arduino Uno*, *NH fuse*, sensor tegangan, sensor arus ACS712, *Ethernet shield*, *RTC DS1307*, *LCD I2C*, dan *router wifi*.



Gambar 3.1 Diagram Fungsional Sistem

Berikut penjelasan mengenai diagram fungsional sistem pada Gambar 3.1 yaitu *Arduino Uno* sebagai mikrokontroler. Pertama-tama saat *NH fuse* diberi input 3 fasa, sensor tegangan akan menerima tegangan yang sudah diubah ke DC oleh *transformator step down* dan rangkaian *signal conditioner* member data berupa sinyal analog menuju ke pin *input* analog *Arduino*. Begitu pula sensor arus ACS712, saat sensor arus dialiri oleh arus sensor akan membaca nilai arus yang diterima dengan mengirim sinyal analog ke *arduino*. Setelah *Arduino* membaca nilai ADC dari masing-masing sensor dan memproses data untuk ditampilkan ke *LCD* dan mengirim data ADC ke *server* melalui *ethernet shield* dan *router*

wifi. Data yang diterima oleh server akan ditampilkan pada program LabVIEW 2016.

3.1 Perancangan *Hardware*

Pada perancangan *hardware* dibagi menjadi beberapa sub bab yang akan dijelaskan per sub bab nya, antara lain :

1. Rangkaian *Shield* Arduino Uno
2. Sensor tegangan dan rangkaian *signal conditioner*
3. Sensor Arus ACS712
4. RTC (*Real Time Clock*) DS1302
5. LCD I2C
6. *Ethernet shield + router*

3.1.1 Perancangan Rangkaian *Shield* Arduino Uno

Pada perancangan rangkaian kontroler ini menggunakan Arduino Uno untuk menerima data dari sensor arus dan tegangan dan akan dikirim melalui *Ethernet shield* dan *router*. Pada rangkaian ini ditambahkan sebuah *shield* untuk sensor tegangan, sensor arus, LCD, dan *real time clock*. Pada *shield* ini terdapat *port* VCC, GND, *port* analog dan *port* digital yang dihubungkan dengan *port* pada arduino. Arduino sebagai mikrokontroler diprogram agar dapat mengontrol, besar nilai tegangan dan arus *NH fuse* serta mengirimnya ke *server* melalui *Ethernet shield* secara *real time*. Berikut ini pada Tabel 3.1 merupakan tabel untuk penggunaan pin pada *shield* Arduino yang akan dibuat :

Tabel 3.1 *Mapping Pin Analog/Digital yang digunakan*

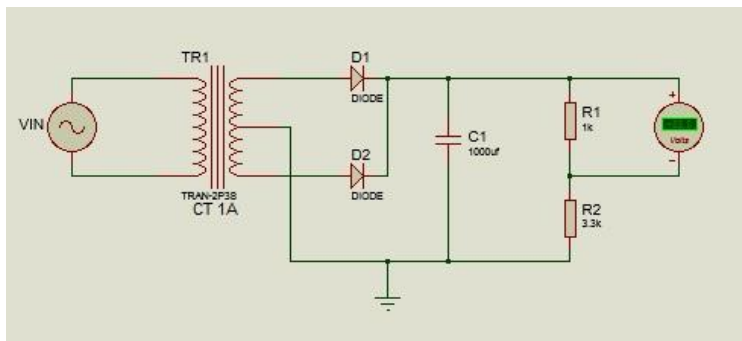
No.	Modul	Pin Analog/Digital
1.	RTC	VCC : 5 Volt GND SDA : SDA SCL : SCL
2.	LCD I2C	VCC : 5 Volt GND SDA : SDA SCL : SCL
3.	Sensor Arus ACS712	VCC :5 Volt GND OUT : Analog 1,2,3



Gambar 3.3 Hasil Pembuatan *Shield* Arduino

3.1.2 Sensor Tegangan

Sensor tegangan merupakan komponen untuk mendeteksi besar tegangan yang masuk di *NH fuse*. Sensor yang dipakai adalah *transformator step down* dengan penyearah gelombang penuh kombinasi rangkaian pembagi tegangan. Rangkaian pembagi tegangan ini menggunakan prinsip hukum *kirchoff* tegangan, memakai resistor yang dipasang secara seri. Skema rangkaian sensor tegangan untuk 1 fasa-nya dapat dilihat pada Gambar 3.4 dan hasil bentuk sensornya dapat dilihat pada Gambar 3.5.



Gambar 3.4 Rangkaian Sensor Tegangan

Pada sensor tegangan Tugas Akhir ini keluaran sensor adalah 0-4 Volt DC. Karena Arduino hanya dapat menerima tegangan DC

antara 0-5 Volt. Untuk mendapatkan sinyal DC menggunakan rangkaian penyearah gelombang penuh ditambah kapasitor sebagai penyetabil tegangan keluaran, sehingga didapat V_{rs} sebesar 12Volt DC dengan ditetapkan R2 sebesar 3,3K Ω . Dengan menggunakan rumus dari persamaan 2.1, didapatkan perhitungan untuk menentukan besar R1 adalah sebagai berikut:

$$4 = \frac{R1}{R1+3,3k} \times 12$$

$$4R1+13,2 = 12 R1$$

$$13,2 = 8 R1$$

$$R1 = 1,6k\Omega$$

(pakai 1k Ω karena di pasaran ada 1 atau 2k Ω , jika menggunakan yang 2k Ω V_{out} yang dikeluarkan lebih dari 4 Volt)

Adapun komponen yang digunakan pada sensor tegangan adalah:

V_{in} : Sumber AC 220Volt (PLN)

TR1 : Trafo CT 1A

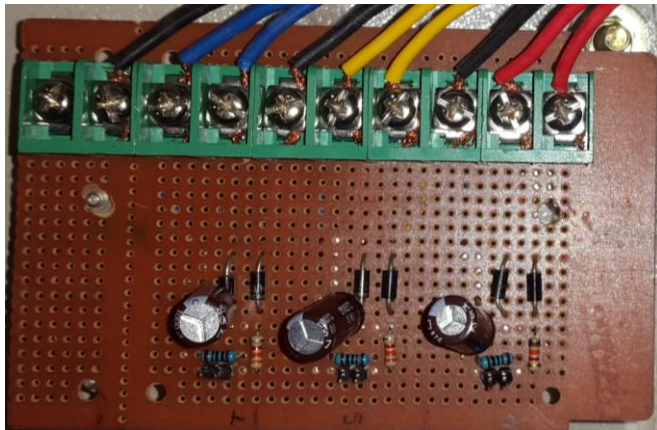
D1, D2: Dioda 1A

C1 : Kapasitor 1000uF

R1 : Resistor 1k Ω

R2 : Resistor 3,3k Ω

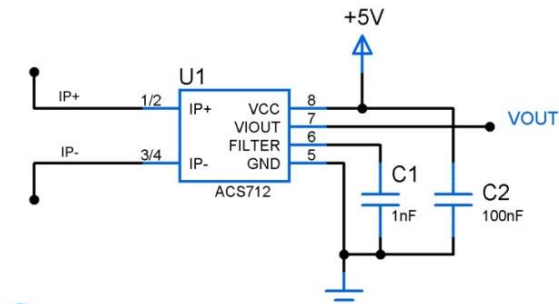
Hasil dari pembuatan rangkaian sensor tegangan dapat dilihat pada Gambar 3.5.



Gambar 3.5 Sensor Tegangan

3.1.3 Sensor Arus

Sensor arus merupakan komponen untuk mendeteksi besar arus beban yang mengalir pada *NH fuse*. Sensor arus yang digunakan adalah sensor arus ACS712 30 Ampere. Sensor ini sudah berbentuk modul yang siap dipakai. Berikut adalah skema rangkaian dari modul ACS712 30 Ampere yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Skema Rangkaian ACS712 30 Ampere

Modul sensor ini membutuhkan tegangan *supply* tegangan sebesar 5 Volt. Keluaran sensor ini adalah 0-5 Volt DC, karena Arduino dapat menerima tegangan 0-5 Volt.

3.1.4 RealTime Clock (RTC) DS1307

Pemakaian RTC dalam alat ini digunakan untuk menampilkan tanggal dan waktu di sistem ini. RTC DS1307 ini memiliki beberapa fitur antara lain:

- Penghitung secara real time untuk detik, menit, jam, hari, tanggal, bulan, dan tahun (valid sampai tahun 2100)
- Format waktu dapat disetting ke dalam format 12 jam (AM/PM) atau 24 jam
- Memiliki kemampuan penyesuaian jumlah hari/bulan terhadap tahun kabisat
- 31 x 8 *Battery-Backed General-Purpose* RAM
- Menggunakan antarmuka i2c (Serial Data dan Serial Clock)
- Menggunakan tegangan *input* kerja 5 VoltDC

- Harus memakai baterai *back-up* dengan range 2-5 VoltDC
- Konsumsi arus pada baterai back-up yaitu hanya 200nA
- *Range* suhu kerja optimal antara -40°C sampai +85°C

RTC DS1307 ini menggunakan 4 pin di antaranya:

- VCC ↔ +5V
- GND ↔ GND
- SCL ↔ SCL
- SDA ↔ SDA

3.1.5 LCD I2C

LCD yang digunakan disini adalah LCD 20x4 warna biru dengan tambahan modul I2C. Modul I2C berguna sekali karena hanya menggunakan 4 pin yaitu pin VCC, GND, SDA, dan SCL. I2C memiliki *address* tersendiri untuk mengakses dengan Arduino. Pada LCD I2C 20x4 ini *address*-nya adalah 0X3F. cara mencari *address* LCD pertama kali kita harus memasukkan program yang bernama I2C *scanner*. Saat menjalankan program ini kita dapat mengetahui *address* LCD I2C yang akan dipakai. Program I2C *scanner* dapat dilihat pada Gambar 3.7.



```

i2c_scanner

#include <Wire.h>

void setup()
{
  //prosedur pembacaan i2c
  Wire.begin();

  Serial.begin(9600); //komunikasi serial menggunakan kecepatan 9600 bps
  while (!Serial);    // Arduino Uno: wait for serial monitor
  Serial.println("\nI2C Scanner");
}

void loop() //fungsi yang berjalan terus menerus
{
  int nDevices;
  byte error, address; //tipe data untuk address dan error yaitu byte
  //ditampilkan di serial monitor "Scanning..."
  Serial.println("Scanning...");

  nDevices = 0;
  for(address = 1; address < 127; address++)
  {
    // The i2c_scanner menggunakan pembalik nilai
    // Write.endTransmission untuk melihat jika perangkat pada alamat dimana
    Wire.beginTransmission(address);
    error = Wire.endTransmission();

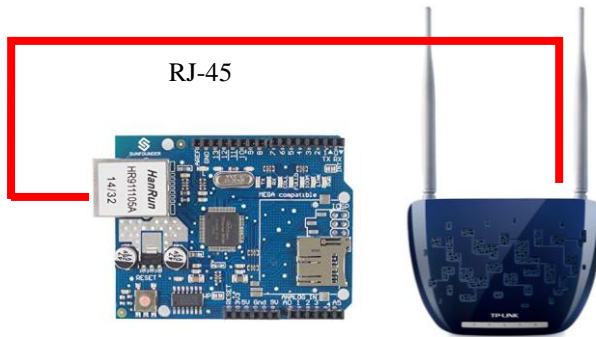
    if (error == 0) //jika sudah tidak ada keerroran
    {
      Serial.print("I2C device found at address 0x");
      //menuliskan di serial monitor "i2c ditemukan"
      if (address<16)
        Serial.print("0");
      Serial.print(address,HEX);
    }
  }
}
  
```

Gambar 3.7 Program I2C Scanner

3.1.6 *Ethernet Shield dan Router Wifi*

Komunikasi antara Arduino dan komputer *server* dilakukan oleh *ethernet shield* dan *router wifi*. Disini *ethernet shield* berfungsi sebagai pengirim data dari Arduino ke *internet*. Saat data sudah masuk di *internet* komputer *server* yang terhubung dengan *wifi router* dapat mengakses data dengan mengeset *IP address* yang sama dengan *ethernet shield*. Program yang digunakan adalah program *web-server* yang sudah ada pada *example* di *software* Arduino IDE dan kita tinggal mengeset *IP* yang akan digunakan untuk konektivitas *wifi*.

Ethernet shield mendapatkan *supply* tegangan dari Arduino langsung. *Router* TP-Link TLWA830 RE disambungkan ke *ethernet shield* menggunakan kabel *RJ-45*. *Router* TP-Link ini mendapat sumber tegangan tersendiri yaitu sumber 9 Volt dari *adaptor* 12 Volt yang disambungkan ke sumber 220 Volt yang terpasang di *LV panel* dengan mengambil sumber salah satu fasa. Konfigurasi antara *ethernet shield* dan *router* dapat dilihat pada Gambar 3.8.



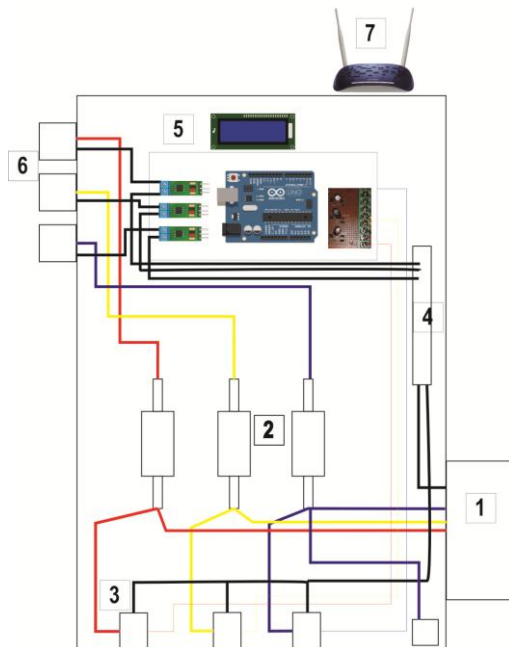
Gambar 3.8 Konfigurasi Antara *Ethernet Shield* dan *Router*

3.1.7 *Panel Prototype*

Perancangan Tugas Akhir ini berupa *prototype*, yaitu pemodelan *NH fuse* pada *LV panel* GTT (Gardu Tiang Trafo). *Prototype* ini bukan merupakan alat yang lengkap, tetapi alat yang perlu dimodifikasi kembali, dikembangkan, dan ditambahkan dengan perangkat atau peralatan yang lain. Seperti hal-nya dengan judul Tugas Akhir ini, pada judul Tugas Akhir ini menggunakan Arduino

Uno dengan media *wifi*. Berbeda dengan judul pendahulunya yang menggunakan mikrokontroller dan media GSM. Hal ini membuktikan bahwa alat ini dapat dikembangkan seterusnya dengan berbagai metode dan cara yang berbeda. *Prototype* ini dimaksudkan untuk mewakili *LV panel* pada GTT yang di dalamnya terdapat seperangkat alat elektronik yang saling bekerjasama.

Box LV panel yang digunakan pada Tugas Akhir ini menggunakan *box* berbahan besi campuran aluminium. Bahan ini sesuai untuk menempatkan perangkat elektronik yang diletakkan secara *out-door*. Ukuran *box panel* adalah 40x20x60 cm (panjang x lebar x tinggi). Ukuran ini telah disesuaikan dengan ukuran atau dimensi peralatan yang digunakan. Ada pula peletakan alat dan *wiring* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Wiring dan Perancangan *Prototype*

Wiring dan peletakan pada panel dilakukan seminimal mungkin agar isi *prototype* tidak terasa kepenuhan. Pada bagian bawah panel yang bernomor 2 ada *NH fuse* dan trafo CT yang ber-

nomor 2 untuk sensor tegangan. Lalu diatas *NH fuse* pada gambar bernomor 2 terdapat Arduino Uno dan beberapa perangkat lain yang dibutuhkan seperti *ethernet shield*, sensor arus, sensor tegangan, RTC, dan LCD 20x4. *Access point* diletakan di luar panel bagian atas seperti yang ditunjukkan pada nomor 7 dengan mengulurkan kabel RJ-45 dan kabel adaptor dari dalam. Pada bagian kiri panel yang bernomor 6 terdapat *output* untuk disambungkan ke beban. Pada bagian kanan panel terdapat MCB dari *input* listrik 3 fasa yang ditandakan dengan nomor 1. Yang bernomor 4 adalah plat tembaga untuk *ground*.

3.2 Perancangan Software

Pada perancangan *software* untuk alat ini berupa pemrograman Arduino Uno di Arduino IDE dan pembuatan tampilan *interface monitoring* di LabVIEW 2016 pada komputer *server*.

3.2.1 Perancangan Program Arduino

Perancangan program pada arduino ini berkaitan dengan pembuatan program untuk sub-sub sistem yang berupa sensor-sensor, *ethernet shield*, RTC, LCD. Langkah awal dalam suatu pemrograman adalah deklarasi variabel dan tipe data yang digunakan. Variabel dan tipe data ini digunakan untuk memudahkan dalam memprogram perangkat apa saja yang ingin diprogram. Untuk sensor pertama adalah mendeklarasi pin *input* A0-A6 untuk sensor arus dan tegangan. Untuk pin A0 s/d A2 digunakan untuk sensor arus ACS712, sedangkan pin A3 s/d A6 digunakan untuk sensor tegangan. Program deklarasi untuk sensor dapat dilihat pada Gambar 3.10.

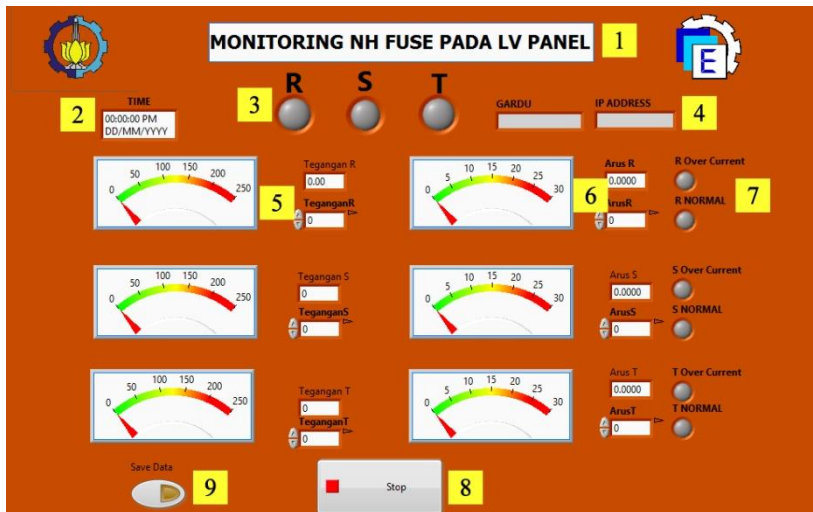
```
int I1=analogRead(A0);  
int I2=analogRead(A1);  
int I3=analogRead(A2);  
int V1=analogRead(A3);  
int V2=analogRead(A4);  
int V3=analogRead(A5);  
  
float ArusR = 0.00;  
float ArusS = 0.00;  
float ArusT = 0.00;  
float TeganganR = 0.00;  
float TeganganS = 0.00;  
float TeganganT = 0.00;
```

Gambar 3.10 Program Deklarasi Pin pada Arduino IDE

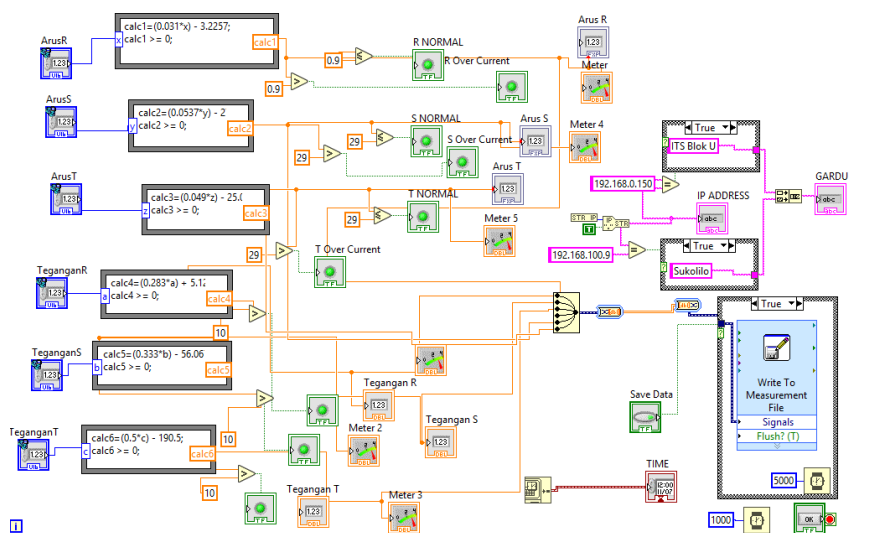
3.2.2 Pembuatan Tampilan *Interface* LabVIEW

Pada pembuatan di *software* LabVIEW ini ada dua tahap yaitu pembuatan *front panel* dan pembuatan *block diagram*. *Front panel* adalah sebuah fitur LabVIEW yang merupakan sebuah tampilan untuk memperlihatkan hasil proses program yang telah disusun dari *block diagram*. *Block diagram* adalah fitur untuk menyusun program yang akan dijalankan.

Untuk program kali ini dibutuhkan *software* tambahan dari *National Instrument* yang dapat membantu mengkoneksikan Arduino dengan LabVIEW melalui *Wifi* untuk menerima data, yaitu NI OPC Server. Data yang diterima dari Arduino akan dihitung kembali di *formula node* dan hasilnya akan tampil di Volt meter dan Ampere meter di *front panel*. Lokasi dari gardu tersebut dapat diketahui dari penyesuaian *server* dengan *IP address* yang tersambung. Tiap gardu memiliki *IP address* yang berbeda. Hasil dari pembuatan desain *front panel* dan *block diagram* dapat dilihat pada Gambar 3.11 dan Gambar 3.12



Gambar 3.11 Desain Tampilan LabVIEW untuk *Prototype*



Gambar 3.12 Block Diagram LabVIEW

Tabel 3.2 Keterangan pada Gambar 3.11

No	Keterangan
1	Judul pada LabVIEW
2	Waktu dan Tanggal
3	Lampu indikator tiap fasa
4	Lokasi gardu dan IP address
5	Volt meter dan indikator nilai ADC sensor tegangan
6	Ampere meter dan indikator nilai ADC sensor tegangan
7	Status arus pada <i>NH fuse</i>
8	Tombol <i>stop</i>
9	Tombol untuk menyimpan data

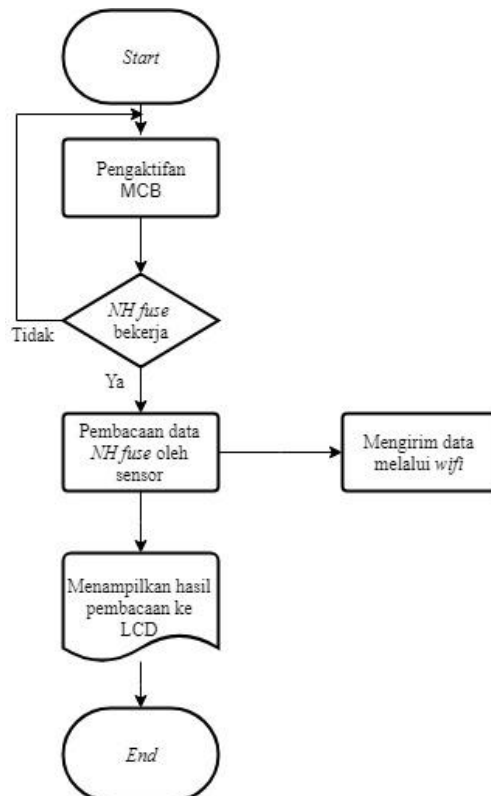
3.2.3 Flowchart Program

Flowchart pada *prototype* ditunjukkan pada Gambar 3.13 dan *flowchat* untuk tampilan komputer *server* ditunjukkan pada Gambar 3.14.

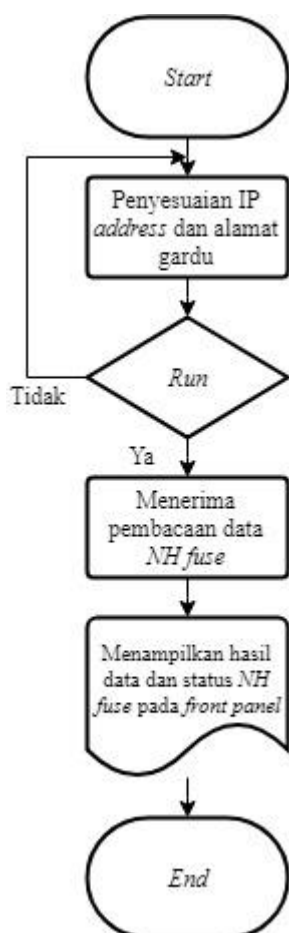
Deskripsi dari *flowchart* pada Gambar 3.13 adalah:

1. MCB diaktifkan dan seluruh alat akan aktif
2. *NH fuse* aktif
3. Arduino membaca ADC dari sensor

4. Data yang diterima Arduino dikirim ke *server* melalui *wifi*
 5. LCD menampilkan data dari sensor
- Deskripsi dari *flowchart* pada Gambar 3.14 adalah:
1. Menjalankan *software* LabVIEW
 2. Dilakukan penyesuaian *IP address* sebelum seluruh program jalan
 3. Saat *IP address* sudah sesuai data dapat diterima oleh LabVIEW
 4. *Front panel* menampilkan data *monitoring NH fuse*



Gambar 3.13 *Flowchart* Sistem pada *Prototype*



Gambar 3.14 Flowchart Sistem pada LabVIEW Server

BAB IV

PENGUJIAN DAN ANALISA DATA

Pada bab ini akan membahas tentang hasil dari pengujian dan analisa atas pembuatan alat untuk “Perancangan Sistem *Monitoring Nh Fuse* Pada *Lv Panel* Berbasis Arduino Dengan Media *Wifi*”. Data pengujian diperlukan untuk implementasi dalam dunia nyata. Kinerja suatu sistem sangat dipengaruhi oleh kinerja tiap bagian dari suatu sistem tersebut.

Untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah terlaksana atau tidak, perlu dilakukan pengujian dan analisa terhadap alat yang dibuat. setelah dilakukan pengujian, dilakukan analisa terhadap bagian-bagian alat yang telah diuji. Kesesuaian sistem dengan perencanaan dapat dilihat dari hasil yang dicapai pada pengujian sistem. Pengujian ini juga bertujuan untuk mencari kelebihan dan kekurangan dari sistem yang telah dibuat. Hasil pengujian tersebut akan dianalisa untuk mengetahui penyebab terjadinya kekurangan atau kesalahan dalam sistem.

4.1 Pengujian Arduino Uno

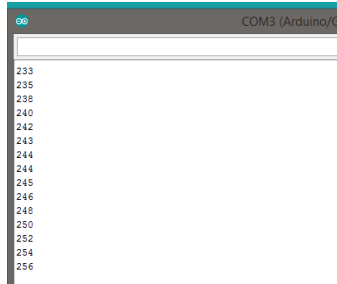
Pengujian pada Arduino Uno bertujuan untuk mengetahui apakah mikrokontroller bekerja baik atau tidak. Pengujian dilakukan dengan memasukkan sebuah program pembacaan data Analog input Arduino. Program untuk percobaan analog *read* bisa dilihat pada Gambar 4.1 berikut :

```
// the setup routine runs once when you press reset:
void setup() {
    // initialize serial communication at 9600 bits per second:
    Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
    // read the input on analog pin 0:
    int sensorValue = analogRead(A0);
    // print out the value you read:
    Serial.println(sensorValue);
    delay(100);          // delay in between reads for stability
}
```

Gambar 4.1 Program *Analog Read* Arduino

Selanjutnya, untuk mengetahui apakah Arduino dapat bekerja bisa dilihat pada *serial monitor*. Hasil dari pengujian ini bisa dilihat pada Gambar 4.2.



Gambar 4.2 Hasil Analog Read Arduino

4.2 Pengujian *Ethernet Shield*

Pengujian *ethernet shield* dilakukan agar mengetahui kerja *ethernet shield* baik atau tidak, karena *ethernet shield* berperan penting dalam pengiriman data ke *server*. Pengujian dilakukan dengan memasukkan program *example webserver* ke dalam Arduino. *Ethernet shield* disambungkan dengan *router* untuk mengetes koneksi dengan *wifi* antara Arduino dengan komputer. Adapun program *webserver* dapat dilihat pada Gambar 4.3.

```
WebServer

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177);

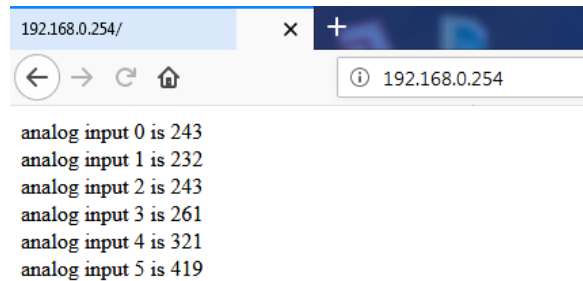
// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
  }

  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}
```

Gambar 4.3 Program Web Server Arduino

Setelah dimasukkan programnya yang dilakukan adalah mengubah *setting* IP pada *setting* wifi setelah tersambung dengan *router*. IP dapat diganti di *properties* wifi pada *setting* internet komputer. setelah tersambung hasil pembacaan analog dapat dilihat pada *browser* komputer dengan menulis IP *address* pada *browser*. Hasil pengujian dapat dilihat pada Gambar 4.4.



Gambar 4.4 Pembacaan Analog *Read* yang Dikirim Melalui *Wifi*

Setelah itu dilakukan pengujian kualitas koneksi antara *access point* dan laptop yang menjadi *server*. Pengujian dilakukan dengan menggunakan *software* Wireshark dengan mengukur *packet loss*, *delay*, dan *throughput*. Adapun standar indeks untuk *packet loss*, *delay*, dan *throughput* dapat dilihat pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3. Standar ini diambil dari TIPHON (*Telecommunications and Internet Protocol Harmonization over Networks*) [12]. Pengujian dilakukan dengan jarak antara *access point* dan PC kurang lebih 20 Meter.

Tabel 4.1 Standar Indeks *Packet Loss* [12]

Kategori	<i>Packet Loss</i>	Indeks
Sangat Bagus	0	4
Bagus	3	3
Sedang	15	2
Buruk	25	1

Tabel 4.2 Standar Indeks *Delay* [12]

Kategori	Besar <i>Delay</i> (ms)	Indeks
Sangat Bagus	<150	4
Bagus	150 s/d 300	3
Sedang	300 s/d 450	2
Buruk	>450	1

Tabel 4.3 Standar Indeks *Throughput* [12]

Kategori <i>Throughput</i>	<i>Throughput</i> (%)	Indeks
Sangat Bagus	100	4
Bagus	75	3
Sedang	50	2
Buruk	<25	1

Pada saat pengujian nilai untuk *packet loss* dapat dilihat pada pojok kanan tampilan Wireshark. Dapat terlihat pada Gambar 4.5 bahwa *packet loss* yang didapat adalah 0,4%. Menurut standard TIPHON nilai ini termasuk baik.

Packets: 35170 · Displayed: 137 (0,4%) · Dropped: 0 (0,0%)	Profile: Defai
--	----------------

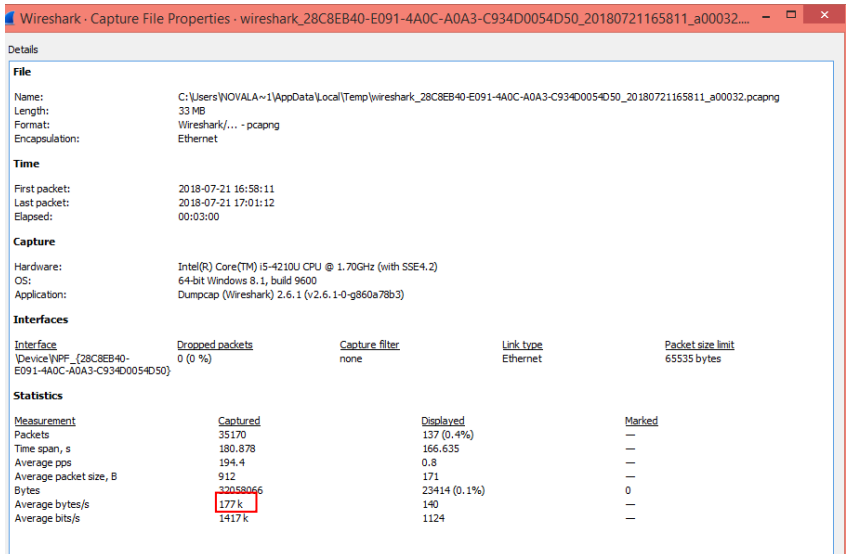
Gambar 4.5 Nilai *Packet Loss* pada Wireshark

Untuk *delay* pengiriman data antara pengirim dan penerima dapat dilihat pada kolom *time delta from previous displayed frame*. Didapatkan data *delay* seperti pada Gambar 4.6.

Protocol	Length	Time delta from previous displayed frame	Identification	Info
TCP	54	0.000088000	0x79e9 (31209)	51759 → 80 [FIN, ACK] Seq=165 Ack=224 Win=64
TCP	54	0.000105000	0x1be0 (7136)	51761 → 443 [ACK] Seq=518 Ack=2713 Win=66048
TCP	54	0.000134000	0x39d2 (14802)	51756 → 5228 [ACK] Seq=979 Ack=300 Win=65792

Gambar 4.6 Nilai *Delay* pada Wireshark

Untuk *throughput* yang didapat dari pengujian Wireshark dapat dilihat pada *capture file properties*. Dapat dilihat pada Gambar 4.7 bahwa nilai *throughput*nya adalah 177Kbytes/s.

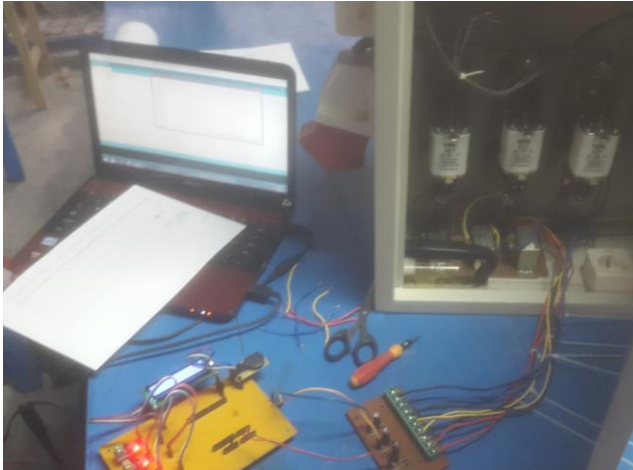


Gambar 4.7 Nilai *Throughput* pada Wireshark

4.3 Pengujian Sensor Tegangan

Pada pengujian sensor tegangan dimana sensor yang dipakai adalah rangkaian pembagi tegangan yang sebelumnya melalui trafo *step down*. Sensor tegangan yang digunakan adalah rangkaian pembagi tegangan yang diperlihatkan pada Gambar 3.5.

Pengujian sensor tegangan menggunakan VariAC milik Laboratorium Elektronika Dasar Departemen Teknik Elektro Otomasi. VariAC dapat memberikan *input* tegangan yang dapat diubah-ubah mulai dari 0-250 Volt. Data yang diambil mengambil range tegangan mulai dari 200 Volt sampai dengan 240 Volt dengan selisih tiap data 1 Volt. Nilai tegangan yang keluar tidak selalu akurat dengan nilai yang diharapkan dikarenakan perputaran knopnya yang susah dan factor dari kondisi VariAC yang sudah lama. Berikut adalah dokumentasi dari pengambilan data untuk sensor tegangan yang dapat dilihat pada Gambar 4.8 dan Gambar 4.10.



Gambar 4.8 Pengambilan Data Sensor Tegangan

```

959
950
941
932
924
918
913
909
908
909
913
919
927
936
946

```

Gambar 4.9 Hasil Pembacaan *Serial Monitor*



Gambar 4.10 VariAC 0-250 Volt

Pengambilan data sensor berupa data tegangan ADC dan ADC yang tampil pada *serial monitor* Arduino IDE seperti pada Gambar 4.9 agar dapat membandingkan data dengan lebih akurat. Data yang diperoleh dapat dilihat pada Tabel 4.4, Tabel 4.5, dan Tabel 4.6.

Tabel 4.4 Hasil Pengukuran Sensor Tegangan Fasa R

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
1	200	199,9	698	3,41
2	201	201,2	704	3,44
3	202	202	703	3,44
4	203	203	705	3,45
5	204	204	707	3,46
6	205	205	704	3,44
7	206	206,2	708	3,46
8	207	207,1	710	3,47
9	208	208,1	714	3,49
10	209	209,1	719	3,51
11	210	210,1	722	3,53
12	211	211,4	725	3,54
13	212	212,1	729	3,56

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
14	213	213,2	732	3,58
15	214	214	735	3,59
16	215	215,2	739	3,61
17	216	216	744	3,64
18	217	217,2	747	3,65
19	218	218	749	3,66
20	219	219,6	755	3,69
21	220	220,1	758	3,70
22	221	221,4	761	3,72
23	222	222,1	763	3,73
24	223	223,2	768	3,75
25	224	224,1	772	3,77
26	225	225	777	3,80
27	226	226,2	780	3,81
28	227	227,3	785	3,84
29	228	228,4	788	3,85
30	229	229,1	791	3,87
31	230	230	795	3,89
32	231	231,2	800	3,91
33	232	232,1	802	3,92
34	233	233	804	3,93
35	234	234,2	810	3,96
36	235	235,2	812	3,97
37	236	236	818	4,00
38	237	237,3	822	4,02
39	238	238,1	828	4,05
40	239	239,3	830	4,06
41	240	240,3	835	4,08

Tabel 4.5 Hasil Pengukuran Sensor Tegangan Fasa S

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
1	200	200,3	743	3,63
2	201	201,3	745	3,64
3	202	202,2	749	3,66
4	203	202,9	751	3,67
5	204	204,1	754	3,69
6	205	205	758	3,70
7	206	206	761	3,72
8	207	207	764	3,73
9	208	208,1	766	3,74
10	209	209,3	771	3,77
11	210	210	773	3,78
12	211	211,1	775	3,79
13	212	212,3	779	3,81
14	213	213,1	782	3,82
15	214	214	783	3,83
16	215	215	786	3,84
17	216	216	789	3,86
18	217	217	792	3,87
19	218	218	795	3,89
20	219	219	798	3,90
21	220	220	801	3,91
22	221	221,2	805	3,93
23	222	222	808	3,95
24	223	223,2	810	3,96
25	224	224	813	3,97
26	225	225	816	3,99
27	226	226,1	820	4,01

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
28	227	227,3	823	4,02
29	228	228,4	826	4,04
30	229	229	829	4,05
31	230	230	831	4,06
32	231	231	833	4,07
33	232	232	837	4,09
34	233	233,3	840	4,11
35	234	234	842	4,12
36	235	234,9	845	4,13
37	236	236	848	4,14
38	237	237	851	4,16
39	238	238	855	4,18
40	239	239	857	4,19
41	240	240	860	4,20

Tabel 4.6 Hasil Pengukuran Sensor Tegangan Fasa T

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
1	200	200,3	733	3,58
2	201	201	736	3,60
3	202	202,3	739	3,61
4	203	203,1	742	3,63
5	204	204,2	745	3,64
6	205	205,1	748	3,66
7	206	206,2	752	3,68
8	207	207	755	3,69
9	208	208,1	759	3,71
10	209	209,1	761	3,72
11	210	210,3	764	3,73

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
12	211	211,3	768	3,75
13	212	212,3	771	3,77
14	213	213	774	3,78
15	214	214	776	3,79
16	215	215,3	778	3,80
17	216	216,1	781	3,82
18	217	217	784	3,83
19	218	218	786	3,84
20	219	219	789	3,86
21	220	220,1	792	3,87
22	221	221,1	795	3,89
23	222	222	798	3,90
24	223	223,2	801	3,91
25	224	224	804	3,93
26	225	225,2	807	3,94
27	226	226	810	3,96
28	227	227,3	812	3,97
29	228	228	815	3,98
30	229	229,3	817	3,99
31	230	230,1	820	4,01
32	231	231,1	823	4,02
33	232	232,1	827	4,04
34	233	233,2	830	4,06
35	234	234,2	833	4,07
36	235	235	836	4,09
37	236	236	838	4,10
38	237	237,3	841	4,11
39	238	238,2	844	4,13

No	Input Tegangan (V)	Tegangan Terukur (V)	ADC	Tegangan ADC (V)
40	239	239	847	4,14
41	240	240,1	850	4,15

Nilai ADC yang diambil disini adalah nilai yang paling banyak keluar atau nilai rata- ratanya. Dari data yang diperoleh bisa disimpulkan bahwa tiap 1 Volt ada perbedaan nilai tegangan ADC antara 0,1-0,3 Volt. Setelah di dapatkan data, dengan menggunakan metode regresi linier *slope intercept* seperti pada Gambar 4.11, Gambar 4.12, dan Gambar 4.13, maka didapat

Fasa R

$$y = 3,5067x - 12,283 \dots\dots\dots(4.1)$$

Fasa S

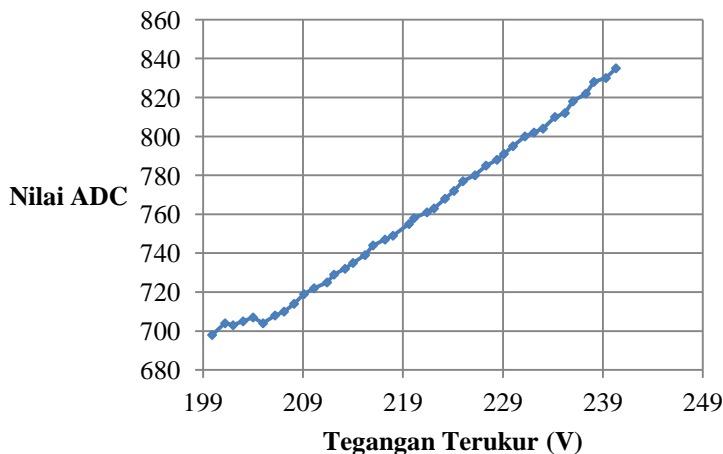
$$y = 2,9386x + 154,84 \dots\dots\dots(4.2)$$

Fasa T

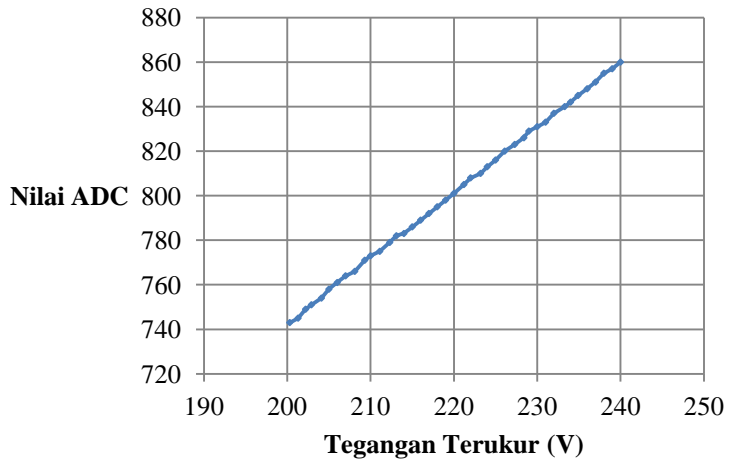
$$y = 2,8914x + 155,75 \dots\dots\dots(4.3)$$

Keterangan : x = Nilai Tegangan (V)

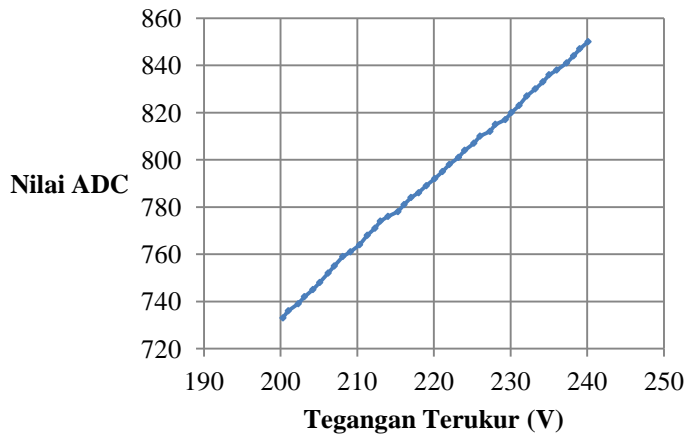
y = Nilai ADC



Gambar 4.11 Data Sensor Tegangan Fasa R



Gambar 4.12 Data Sensor Tegangan Fasa S



Gambar 4.13 Data Sensor Tegangan Fasa T

Dari data yang didapat bahwa hasil nilai tegangan ADC tiap sensor linier terhadap *input* tegangan. Selanjutnya, untuk mengetahui sensor dianggap linier, maka dilakukan pengambilan data ulang

dengan memasukkan hasil perhitungan *slope intercept* ke program Arduino. Setelah didapatkan data ulang dan mendapat data tegangan yang terbaca, selanjutnya mencari % *error* yang ada di setiap sensor. Didapatkan data sebagai berikut:

$$\%Error = \frac{\text{Hasil multimeter} - \text{Tampil LCD}}{\text{Hasil multimeter}} \times 100\% \dots\dots\dots(4.4)$$

Tabel 4.7 Data Perbandingan Tegangan Fasa R

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
1	200	199,83	200	0,085
2	201	200,11	201,1	0,492292
3	202	201,81	202,4	0,291502
4	203	202,66	203,1	0,216642
5	204	203,51	204,2	0,337904
6	205	204,36	205,2	0,409357
7	206	205,49	206	0,247573
8	207	206,62	207	0,183575
9	208	207,76	208	0,115385
10	209	208,6	209,2	0,286807
11	210	209,74	210,3	0,266286
12	211	210,87	211,2	0,15625
13	212	211,15	212	0,400943
14	213	212,57	213	0,201878
15	214	213,42	214,1	0,317609
16	215	214,55	215	0,209302
17	216	215,96	216	0,018519
18	217	217,09	217,2	0,050645
19	218	217,94	218,1	0,073361
20	219	218,79	219,2	0,187044
21	220	219,92	220,2	0,127157
22	221	220,49	221	0,230769

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
23	222	221,91	222,1	0,085547
24	223	223	223,3	0,134348
25	224	223,89	224	0,049107
26	225	225,02	225,3	0,124279
27	226	225,87	226	0,057522
28	227	226,6	227	0,176211
29	228	228,13	228,2	0,030675
30	229	228,9	229	0,043668
31	230	230,11	230,3	0,082501
32	231	231,24	231,4	0,069144
33	232	231,81	232,1	0,124946
34	233	233,23	233,3	0,030004
35	234	234,36	234,4	0,017065
36	235	234,92	235	0,034043
37	236	235,77	236	0,097458
38	237	236,13	237,2	0,451096
39	238	238,89	238,2	0,28967
40	239	239,17	239,2	0,012542
41	240	240,87	240,2	0,27893

Tabel 4.8 Data Perbandingan Tegangan Fasa S

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
1	200	199,48	200	0,26
2	201	200,84	201,2	0,1789
3	202	201,86	202,2	0,16815
4	203	202,88	203	0,059113
5	204	203,9	204,2	0,146915
6	205	204,92	205	0,039024

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
7	206	205,94	206,1	0,077632
8	207	206,96	207,4	0,21215
9	208	208,66	209,1	0,210426
10	209	209,69	210,1	0,195145
11	210	211,05	211,3	0,118315
12	211	211,73	212,1	0,174446
13	212	212,75	212,1	0,30646
14	213	212,75	213	0,117371
15	214	213,77	214,2	0,200747
16	215	214,79	215	0,097674
17	216	215,81	216	0,087963
18	217	216,83	217,2	0,17035
19	218	217,85	218,2	0,160403
20	219	218,87	220	0,513636
21	220	219,87	219,3	0,25992
22	221	220,91	221,3	0,176231
23	222	221,93	222,3	0,166442
24	223	222,95	223	0,022422
25	224	223,97	224	0,013393
26	225	224,99	225	0,004444
27	226	226	226,2	0,088417
28	227	226,7	227	0,132159
29	228	227,72	228	0,122807
30	229	228,74	229,2	0,200698
31	230	229,76	230,3	0,234477
32	231	230,78	231,2	0,181661
33	232	231,8	232	0,086207
34	233	232,82	233	0,077253

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
35	234	233,84	234	0,068376
36	235	235,2	235,3	0,042499
37	236	235,88	236	0,050847
38	237	236,9	237,2	0,126476
39	238	237,92	238	0,033613
40	239	238,94	239	0,025105
41	240	239,96	240	0,016667

Tabel 4.9 Data Perbandingan Tegangan Fasa T

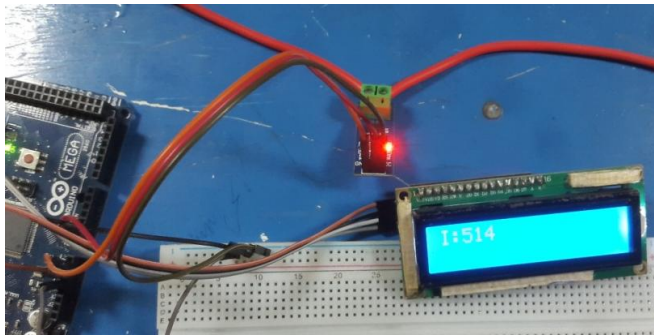
No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
1	200	199,66	199,99	0,165008
2	201	200,69	201	0,154229
3	202	201,73	202,1	0,183078
4	203	202,77	203	0,1133
5	204	204,15	204,3	0,073421
6	205	205,19	205,3	0,05358
7	206	205,88	206	0,058252
8	207	206,91	207,3	0,188133
9	208	207,95	208	0,024038
10	209	208,99	209	0,004785
11	210	210,02	210,2	0,085633
12	211	211,06	211,1	0,018948
13	212	212,1	212,2	0,047125
14	213	212,79	213	0,098592
15	214	213,83	214,1	0,126109
16	215	214,86	215	0,065116
17	216	215,9	216	0,046296
18	217	216,94	217	0,02765

No	Input Tegangan (V)	Tegangan Terukur (V)	Tegangan Hasil (V)	% Error
19	218	217,97	218	0,013761
20	219	219,01	219,2	0,086679
21	220	220,05	220,2	0,06812
22	221	220,74	221	0,117647
23	222	222,12	222,2	0,036004
24	223	222,81	223	0,085202
25	224	223,85	224	0,066964
26	225	224,89	225	0,048889
27	226	225,93	226	0,030973
28	227	226,96	227	0,017621
29	228	228	228,2	0,087642
30	229	229,03	229,2	0,074171
31	230	229,07	230,2	0,490877
32	231	231,1	231,1	0
33	232	231,14	231,9	0,327727
34	233	233,18	233,3	0,051436
35	234	234,2	234,3	0,04268
36	235	235,25	235,2	0,02126
37	236	235,94	236	0,025424
38	237	236,98	237,2	0,092749
39	238	238,02	238,1	0,033599
40	239	239,4	239,3	0,04179
41	240	240,09	240,1	0,004165

Dari hasil pengujian didapatkan data data % *error* seperti yang diperlihatkan pada Tabel 4.7, Tabel 4.8, dan Tabel 4.9. *Error* yang didapat untuk fasa R sekitar 0,01% sampai dengan 0,45%. Untuk fasa S % *error* yang didapat sekitar 0,01% sampai dengan 0,5%. Untuk fasa T didapatkan % *error* sekitar 0% sampai dengan 0,49%.

4.3 Pengujian Sensor Arus ACS712

Pengujian sensor arus bertujuan agar nilai arus yang dibaca oleh sensor dapat akurat. Pengujian dilakukan dengan memberi beban berupa lampu pijar dari 25 Watt hingga 125 Watt. Dari perlakuan tersebut didapatkan nilai ADC yang dihasilkan oleh sensor ACS712. Pengujian dilakukan dengan pemasangan ACS712 dengan Arduino dan membaca nilai ADC di *serial monitor* dan LCD. Dari nilai ADC yang diperoleh, data tersebut dibandingkan dengan nilai arus yang dibaca oleh *clampmeter*. Pada Gambar 4.14 merupakan dokumentasi dari pengambilan data untuk sensor arus ACS712. Gambar 4.15 adalah hasil yang ditampilkan pada *serial monitor*. Hasil data dari pengujian tiga sensor arus tiap fasa dapat dilihat pada Tabel 4.10, Tabel 4.11, dan Tabel 4.12.



Gambar 4.14 Pengambilan Data Sensor ACS712

```
514
514
514
514
513
513
514
514
514
514
514
514
...
```

Gambar 4.15 Hasil Pembacaan *Serial Monitor* ACS712

Tabel 4.10 Data Sensor Arus ACS712 Fasa R

No	Beban (Watt)	ADC	Tegangan ADC (V)	Arus Terukur (A)
1	0	104	0,552	0
2	25	107	0,572	0,11
3	50	112	0,582	0,22
4	75	116	0,601	0,33
5	100	119	0,635	0,43
6	125	121	0,655	0,55
7	150	125	0,679	0,66

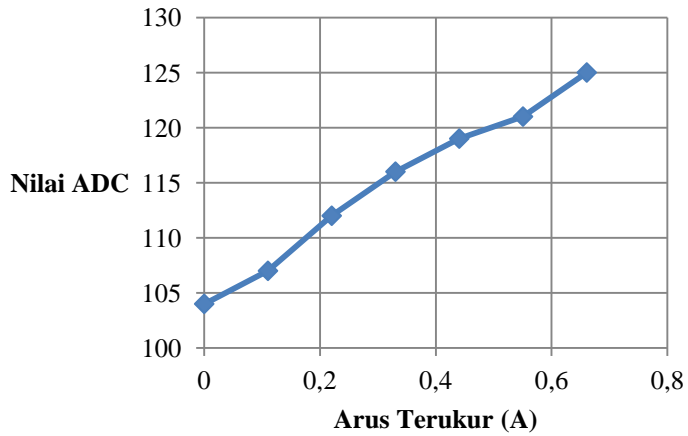
Tabel 4.11 Data Sensor Arus ACS712 Fasa S

No	Beban (Watt)	ADC	Tegangan ADC (V)	Arus Terukur (A)
1	0	511	2,439	0
2	25	514	2,454	0,109
3	50	516	2,463	0,218
4	75	518	2,473	0,327
5	100	520	2,483	0,437
6	125	522	2,498	0,546
7	150	524	2,507	0,655

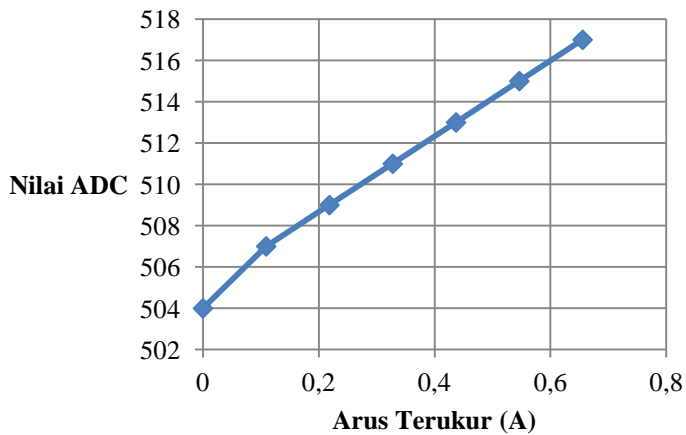
Tabel 4.12 Data Sensor Arus ACS712 Fasa T

No	Beban (Watt)	ADC	Tegangan ADC (V)	Arus Terukur (A)
1	0	515	2,473	0
2	25	517	2,483	0,109
3	50	519	2,493	0,219
4	75	521	2,502	0,329
5	100	524	2,512	0,439
6	125	527	2,522	0,549
7	150	529	2,532	0,659

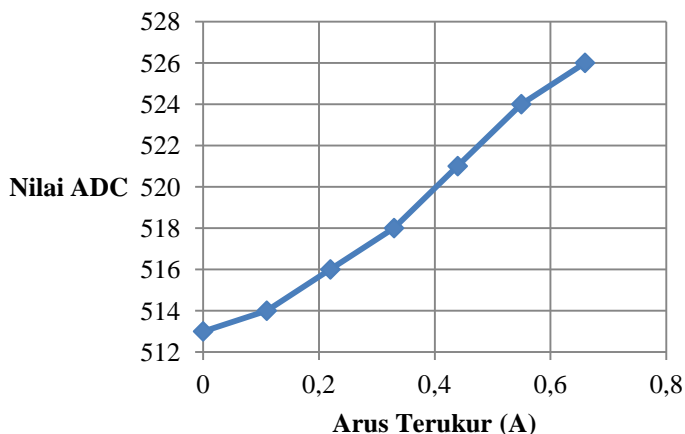
Setelah mendapatkan data ADC dari *output* sensor dilakukan linearisasi data ADC yang ditunjukkan pada Gambar 4.16, 4.17, dan 4.19.



Gambar 4.16 Data Linearisasi Sensor Arus Fasa R



Gambar 4.17 Data Linearisasi Sensor Arus Fasa S



Gambar 4.18 Data Linearisasi Sensor Arus Fasa T

Setelah di dapatkan data tersebut, dengan menggunakan metode regresi linier *slope intercept*, maka diperoleh Persamaan tiap sensornya, yaitu sebagai berikut:

$$\text{Fasa R} \quad y = 31,794x + 104,36 \dots\dots\dots(4.5)$$

$$\text{Fasa S} \quad y = 19,285x + 504,54 \dots\dots\dots(4.6)$$

$$\text{Fasa T} \quad y = 20,8x + 512 \dots\dots\dots(4.7)$$

Keterangan : x = Arus (I)

y = Nilai ADC

Selanjutnya, untuk mengetahui sensor dianggap linier, maka dilakukan pengambilan data ulang dengan memasukkan hasil perhitungan *slope intercept* ke program Arduino.

Setelah didapatkan data ulang dan mendapat data tegangan yang terbaca, selanjutnya mencari % *error* dengan rumus pada persamaan 4.4 di setiap sensor. Didapatkan data seperti yang tertera pada Tabel 4.13, Tabel 4.14, Tabel 4.15

Tabel 4.13 Data Perbandingan Sensor Arus Fasa R

No	Beban (Watt)	Hasil Arus (A)	Arus Terukur (A)	%Error
1	0	0	0	0

No	Beban (Watt)	Hasil Arus (A)	Arus Terukur (A)	%Error
2	25	0,08	0,11	22,241
3	50	0,24	0,22	9,508
4	75	0,36	0,33	10,673
5	100	0,45	0,43	4,193
6	125	0,52	0,55	5,345
7	150	0,64	0,66	2,286

Tabel 4.14 Data Perbandingan Sensor Arus Fasa S

No	Beban (Watt)	Hasil Arus (A)	Arus Terukur (A)	%Error
1	0	0,01	0	0
2	25	0,11	0,109	2,87
3	50	0,22	0,218	1,34
4	75	0,32	0,327	2,74
5	100	0,42	0,437	3,45
6	125	0,53	0,546	3,87
7	150	0,63	0,655	4,15

Tabel 4.15 Data Perbandingan Sensor Arus Fasa T

No	Beban (Watt)	Hasil Arus (A)	Arus Terukur (A)	%Error
1	0	0,01	0	0,28
2	25	0,11	0,109	8,36
3	50	0,20	0,219	11,24
4	75	0,29	0,329	2,31
5	100	0,42	0,439	3,04
6	125	0,56	0,549	0,29
7	150	0,65	0,659	0,28

Dari hasil pengujian didapatkan data data % error seperti yang diperlihatkan pada Tabel 4.13, Tabel 4.14, dan Tabel 4.15. Er-

ror yang didapat untuk arus fasa R sekitar 0% sampai dengan 22%. Untuk arus fasa S % *error* yang didapat sekitar 0% sampai dengan 4%. Untuk fasa T didapatkan % *error* sekitar 0,28% sampai dengan 11,24%.

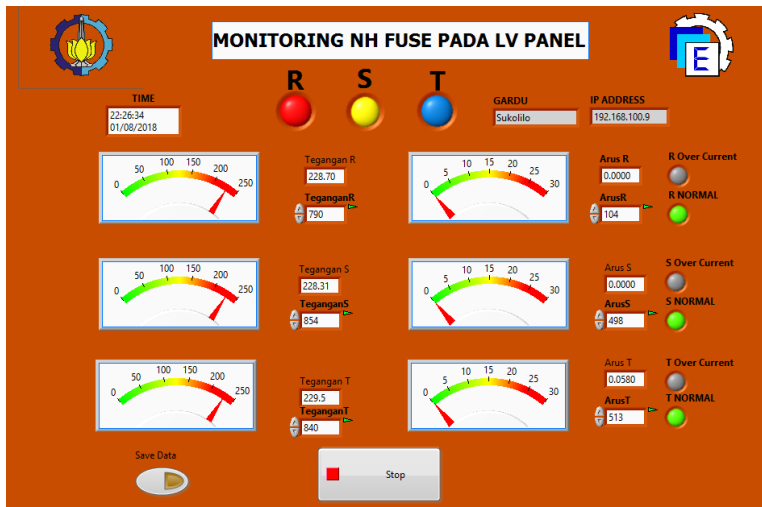
4.4 Pengujian *Software* LabVIEW

Setelah pengujian antara Arduino, *ethernet shield*, dan sensor-sensor perlu diadakan pengujian *Software LabVIEW* untuk mengetahui apakah Arduino dan *ethernet shield* dapat mengirimkan data pada *software* LabVIEW atau tidak.

Pengujian dilakukan di Laboratorium Elektronika Dasar dengan menggunakan sumber 3 fasa. Pengujian dilakukan dengan mengetes koneksi, nilai ADC yang masuk, tegangan serta arus. Gambar 4.19 adalah kondisi *prototype* saat dinyalakan untuk menguji *software*. Gambar 4.20 adalah tampilan saat pengujian *monitoring* dengan LabVIEW berhasil dilakukan.



Gambar 4.19 Penyalan *Prorotype* Saat Menguji *Software*



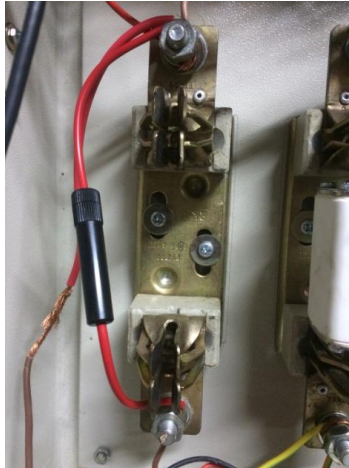
Gambar 4.20 Pengujian *Software* LabVIEW

4.5 Pengujian Keseluruhan

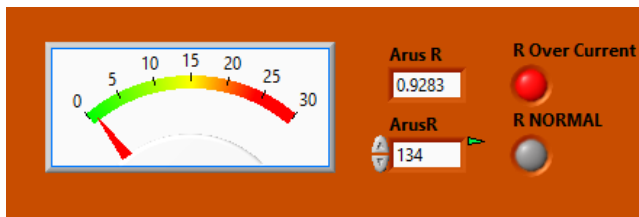
Setelah dilakukan pengujian untuk LabVIEW, selanjutnya dilakukan pengujian keseluruhan. Disini Pengujian keseluruhan juga meliputi simulasi saat kondisi beban lebih. Situasi beban lebih dilakukan dengan menggunakan simulasi. Pada simulasi ini dilakukan dengan mengganti *NH fuse* dengan sekring 1 Ampere. Pengujian dilakukan dengan menggunakan sekring 1 Ampere karena *NH fuse* yang ada disini adalah 50 Ampere. Karena tidak ada beban yang dapat mencapai arus hingga 50 Ampere maka *NH fuse* diganti dengan sekring 1 Ampere. Saat program dijalankan kita juga bisa langsung tahu lokasi gardu mana yang akan di-*monitoring* saat itu juga.

Prinsipnya sama, sekring 1 Ampere akan putus saat arus yang mengalir mencapai 1 ampere. Di pengujian ini menggunakan beban sebesar 200 – 250 Watt. *NH fuse* yang diganti dengan sekring adalah *NH fuse* pada fasa R. Pada Gambar 4.21 adalah pemasangan sekring untuk pengganti *NH fuse*. Pada LabVIEW juga diberi indikator saat mendekati batas maksimal *fuse*. Indikator ini akan menyala saat arus melebihi angka 0,9 Ampere seperti yang ditunjukkan pada Gambar 4.22. Untuk memberi tahu saat *fuse* sudah putus lampu indikator status *NH fuse* fasa R pada LabVIEW akan

mati seperti yang ditunjukkan pada Gambar 4.23. Pada kolom lokasi di tampilan LabVIEW juga akan memberi tahu lokasi *NH fuse* yang di-monitoring seperti pada Gambar 4.24.



Gambar 4.21 Pemasangan Sekring pada Salah Satu Fasa



Gambar 4.22 Tampilan Indikator Arus dan Status Arus pada *NH Fuse* Saat Arus Mendekati Batas



Gambar 4.23 Tampilan Indikator Status *NH Fuse* LabVIEW Saat Fasa R Putus



Gambar 4.24 Tampilan Lokasi Gardu dan IP Address Saat Program Berjalan

Adapun data yang didapat saat terjadinya beban lebih dan sekring putus dapat dilihat pada Tabel 4.16. Dari data yang didapat saat sekring putus, nilai yang dibaca sensor tegangan fasa R mengalami penurunan begitu pula sensor arus.

Tabel 4.16 Data Pengujian Keseluruhan

Tanggal Waktu	Tegangan R (V)	Tegangan S (V)	Tegangan T (V)	Arus R (A)	Arus S (A)	Arus T (A)
11/7/2018 19:05:27	225,87	226,32	226,00	0,28	0,01	0,06
11/7/2018 19:05:33	225,87	226,65	226,50	0,56	0,01	0,06
11/7/2018 19:05:39	225,87	226,65	226,50	0,88	0,01	0,06
11/7/2018 19:05:44	225,87	226,98	226,00	0,99	0,01	0,06
11/7/2018 19:05:50	77,29	226,32	226,00	0,03	0,01	0,06
11/7/2018 19:05:55	28,90	226,65	226,50	0,00	0,02	0,06
11/7/2018 19:06:01	13,90	226,32	226,50	0,00	0,02	0,01
11/7/2018 19:06:07	8,81	226,32	226,00	0,03	0,01	0,01

Tanggal Waktu	Tegan-gan R (V)	Tegan-gan S (V)	Tegan-gan T (V)	Arus R (A)	Arus S (A)	Arus T (A)
11/7/2018 19:06:12	6,83	226,32	226,00	0,03	0,01	0,06
11/7/2018 19:06:18	5,98	226,32	225,50	0,00	0,01	0,06
11/7/2018 19:06:23	5,41	226,65	225,50	0,03	0,01	0,06

BAB V

PENUTUP

5.1 Kesimpulan

Pada penelitian kali ini, dapat diambil kesimpulan antara lain sebagai berikut:

1. *Monitoring NH fuse* pada LabVIEW dapat memberikan info lokasi gardu dan apabila arus pada *NH fuse* mulai mendekati batas indikator *over current akan menyala* dan saat *NH fuse* putus indikator status akan mati.
2. Nilai dari sensor tegangan pada fasa R linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 0,1453%.
3. Nilai dari sensor tegangan pada fasa S linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 0,2138%.
4. Nilai dari sensor tegangan pada fasa R linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 0,0798%.
5. Nilai dari sensor arus pada fasa R linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 9,041%.
6. Nilai dari sensor arus pada fasa S linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 3,07%.
7. Nilai dari sensor arus pada fasa T linier terhadap perubahan tegangan dengan presentase rata – rata *error* sensor sebesar 4,19%.
8. *Ethernet shield* dan *access point* yang digunakan memiliki kemampuan *throughput* hingga 177Kbyte/s, dengan *delay* 0,15ms, dan *packet loss* 0,4%

5.2 Saran

Saran untuk penelitian selanjutnya yaitu diharapkan dapat menggunakan perangkat sensor dengan kelinieran yang lebih baik. Lalu disarankan untuk menambahkan *supply* tambahan untuk modul atau *shield* Arduino dan *router wifi* yang digunakan agar saat terjadi gangguan Arduino dan *router* masih berfungsi dan dapat mengirim data ke *server*.

-----Halaman ini sengaja dikosongkan-----

DAFTAR PUSTAKA

- [1] Farizka, Elviena, “Perancangan Sistem *Monitoring* Kerja *NH Fuse* pada *LV Panel* Menggunakan Mikrokontroler dengan Media Modem GSM”, ***Tugas Akhir***, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2013
- [2] Moch Fajar Wicaksono dkk, ***Mudah Belajar Mikrokontroller Arduino***, Penerbit Informatika, Bandung, 2017
- [3]***LCD I2C***, pembahasan tentan LCD dan I2C, <https://mikroavr.com/arduino-lcd-i2c/>
- [4] Farrah Fadilah, “Telemetering Kebocoran Pipa pada Distriusi Air dengan Komunikasi Ethernet”, ***Tugas Akhir***, Program D3 Teknik Elektro FTI-ITS, Surabaya, 2017
- [5]***Sensor Arus ACS712***, pembahasan tentang sensor arus acs 712, <https://depokinstruments.com/tag/belajar-sensor-arus-listrik/> (diakses pada tanggal 14 Mei 2018)
- [6]***Rangkaian Pembagi Tegangan***, pembahasan tentang rangkaian pembagi tegangan, <http://instrumentasi.lecture.ub.ac.id/rangkaian-pembagi-tegangan> (diakses pada tanggal 3 Juni 2018)
- [7]***Access Point TP Link WA830RE***, Penjelasan dan spesifikasi, https://www.tp-link.com/id/products/details/cat-10_TL-WA830RE.html
- [8] D.E.Johannson, “Telemetering Application and Operation on Large System”, ***AIEE Paper no : 60-242***, 1960
- [9] Herman Dwi Surjono, ***Elektronika Teori dan Penerapan***, Cerdas Ulet Kreatif, Jember, 2011
- [10] Wahyudi Sarimun, ***Proteksi Sistem Distribusi Tenaga Listrik***, Garamond, Depok, 2012
- [11]***Perangkat Hubung Bagi***, SPLN 118-3-1:1996 Standar Tentang Perangkat Hubung Bagi Tegangan Rendah Gardu Distribusi, PT. PLN (Persero), Jakarta
- [12]***Quality of Service***, NN, TIPHON. *Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON General Aspects of Quality of Service (QoS))*. 2002.

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1. Program Arduino

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SPI.h>
#include <Ethernet.h>
#include <Modbus.h>
#include <ModbusIP.h>
#include <DS1307.h>
#include <SD.h>

DS1307 rtc;

LiquidCrystal_I2C lcd(0x3F,20,4);
const int sencalc = 100;
const int sencalc1 = 101;
const int sencalc2 = 102;
const int sencalc3 = 103;
const int sencalc4 = 104;
const int sencalc5 = 105;

const int sensorPin1 = A0;
const int sensorPin2 = A1;
const int sensorPin3 = A2;
const int sensorPin4 = A3;
const int sensorPin5 = A4;
const int sensorPin6 = A5;

ModbusIP mb;

long ts;

const int chipSelect = 4;

//int Calc;
```

```

void setup() {
  byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
  byte ip[] = { 192, 168, 0, 200 };
  mb.config(mac, ip);

  //rtc.set(10, 29, 23, 2, 7, 2018);

  lcd.init();           // initialize the lcd

  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(5,1);
  lcd.print("MONITORING");
  lcd.setCursor(7,2);
  lcd.print("NH FUSE");
  delay(4000);
  lcd.clear ();
  Serial.begin(9600);

  Serial.println("Init RTC...");
  lcd.setCursor(0,0);
  lcd.print("Init RTC...");

  mb.addIreg(sencalc);
  mb.addIreg(sencalc1);
  mb.addIreg(sencalc2);
  mb.addIreg(sencalc3);
  mb.addIreg(sencalc4);
  mb.addIreg(sencalc5);

  rtc.start();

  Serial.print("Initializing SD card...");
  lcd.setCursor(0,1);
  lcd.print("Initializing SD card...");

```

```

// see if the card is present and can be initialized:
if (!SD.begin(chipSelect)) {
  Serial.println("Card failed, or not present");
  lcd.setCursor(0,2);
  lcd.print("Card failed, or not present");
  delay(3000);
  // don't do anything more:
  return;
}
Serial.println("card initialized.");
lcd.setCursor(0,3);
lcd.print("card initialized.");

delay(3000);
lcd.clear ();

ts = millis();

}

void loop() {

  mb.task();

  if (millis() > ts + 1000)

  {
    //cli();

    int I1=analogRead(A0);
    int I2=analogRead(A1);
    int I3=analogRead(A2);
    int V1=analogRead(A3);
    int V2=analogRead(A4);
    int V3=analogRead(A5);
  }
}

```

```

float ArusR = 0.00;
float ArusS = 0.00;
float ArusT = 0.00;
float TeganganR = 0.00;
float TeganganS = 0.00;
float TeganganT = 0.00;

TeganganR = (0.283*V1) + 5.1274;
TeganganS = (0.3402*V2) - 52.609;
TeganganT = (0.3456*V3) - 53.668;
ArusR = (0.0224*I1) - 2.4955;
ArusS = (0.0436*I2) - 21.758;
ArusT = (0.05*I3) - 25.3;

mb.Ireg(sencalc, I1);
mb.Ireg(sencalc1, I2);
mb.Ireg(sencalc2, I3);
mb.Ireg(sencalc3, V1);
mb.Ireg(sencalc4, V2);
mb.Ireg(sencalc5, V3);

Serial.print (ArusR);
Serial.println (" I ");
Serial.print (ArusS);
Serial.println (" I ");
Serial.print (ArusT);
Serial.println (" I ");
Serial.print (TeganganR);
Serial.println(" V ");
Serial.print (TeganganS);
Serial.println(" V ");
Serial.print (TeganganT);
Serial.println(" V ");

lcd.setCursor(0,0);
lcd.print("V1:");
lcd.setCursor(3,0);
lcd.print(TeganganR);
lcd.println("V ");

```



```
lcd.setCursor(11,0);  
lcd.print("I1:");  
lcd.print(ArusR);  
lcd.print("A");
```

```
lcd.setCursor(0,1);  
lcd.print("V2:");  
lcd.setCursor(3,1);  
lcd.print(TeganganS);  
lcd.println("V ");
```

```
lcd.setCursor(11,1);  
lcd.print("I2:");  
lcd.print(ArusS);  
lcd.print("A");
```

```
lcd.setCursor(0,2);  
lcd.print("V3:");  
lcd.setCursor(3,2);  
lcd.print(TeganganT);  
lcd.println("V ");
```

```
lcd.setCursor(11,2);  
lcd.print("I3:");  
lcd.print(ArusT);  
lcd.print("A");
```

```
RTC();
```

```
uint8_t sec, min, hour, day, month;  
uint16_t year;
```

```
rtc.get(&sec, &min, &hour, &day, &month, &year);
```

```
File dataFile = SD.open("datalog.txt", FILE_WRITE);  
if (dataFile){  
dataFile.print(day, DEC);  
dataFile.print("/");
```

```

dataFile.print(month, DEC);
dataFile.print("/");
dataFile.print(year, DEC);
dataFile.print(" ");
dataFile.print(hour, DEC);
dataFile.print(":");
dataFile.print(min, DEC);
dataFile.print(":");
dataFile.print(sec, DEC);
dataFile.print(" ");
dataFile.print(TeganganR);
dataFile.print(" ");
dataFile.print(TeganganS);
dataFile.print(" ");
dataFile.print(TeganganT);
dataFile.print(" ");
dataFile.print(ArusR);
dataFile.print(" ");
dataFile.print(ArusS);
dataFile.print(" ");
dataFile.print(ArusT);
dataFile.println("");
    dataFile.close();}

}
}
void RTC()
{
    uint8_t sec, min, hour, day, month;
    uint16_t year;

    //get time from RTC
    rtc.get(&sec, &min, &hour, &day, &month, &year);

    Serial.print("\nDate: ");
    Serial.print(day, DEC);
    Serial.print(".");
    Serial.print(month, DEC);
    Serial.print(".");

```

```

Serial.print(year, DEC);

lcd.setCursor(0, 3);
//lcd.print("Tgl: ");
//lcd.print("\nDate: ");
lcd.print(day, DEC);
lcd.print("/");
lcd.print(month, DEC);
lcd.print("/");
lcd.print(year, DEC);

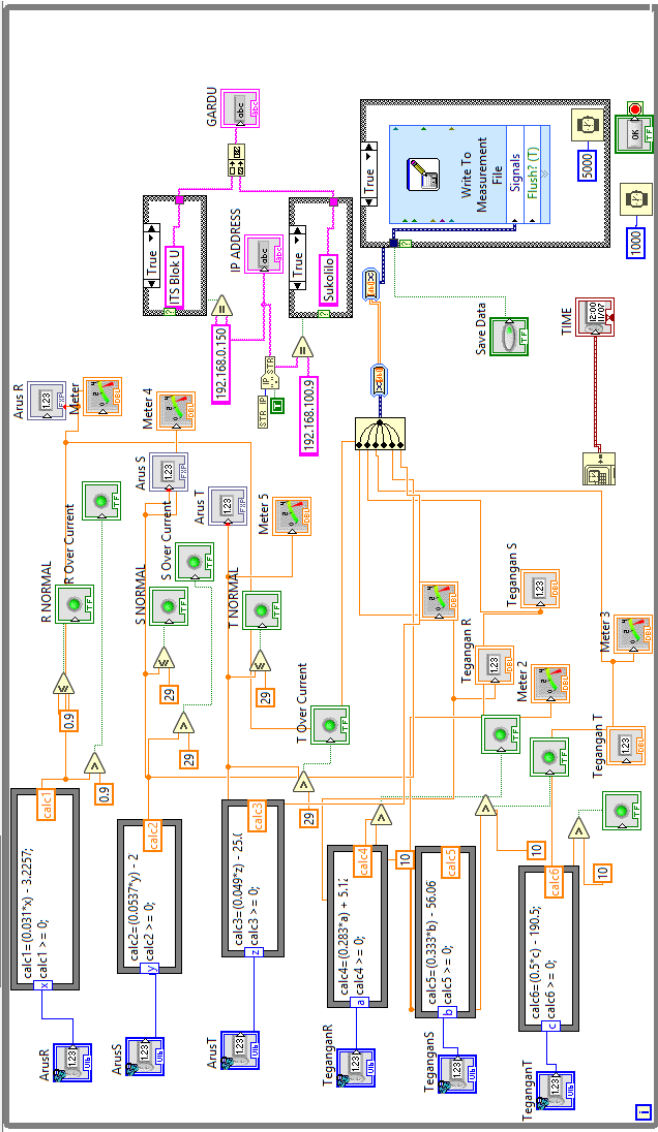
Serial.print("\nTime: ");
Serial.print(hour, DEC);
Serial.print(":");
Serial.print(min, DEC);
Serial.print(":");
Serial.print(sec, DEC);

lcd.setCursor(10, 3);
//lcd.print("Jam: ");
//lcd.print("\nTime: ");
lcd.print(hour, DEC);
lcd.print(":");
lcd.print(min, DEC);
lcd.print(":");
lcd.print(sec, DEC);

}

```

A.2. Program LabVIEW



LAMPIRAN B

B.1 Datasheet Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

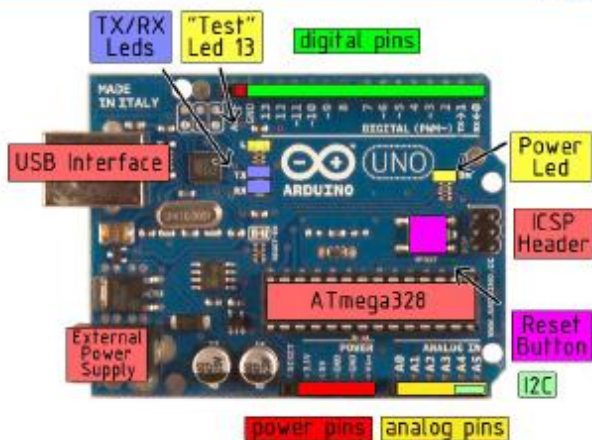


EAGLE file: [arduino-duino-uno-schematic.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



radiospares

RADIONICS



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with `analogReference()`.
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The 8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an ".inf" file is required.

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I²C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I²C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the Tools > Board menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



radiospares

RADIONICS



B.2 Datasheet *Ethernet Shield*

Arduino Ethernet Shield



Download: [arduino-ethernet-shield-05-schematic.pdf](#), [arduino-ethernet-shield-05-reference-design.zip](#)

Download: [arduino-ethernet-shield-schematic.pdf](#), [arduino-ethernet-shield-reference-design.zip](#)

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the [Wiznet W5100](#) ethernet chip ([datasheet](#)). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the [Ethernet library](#) to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the [sdfatlib](#) by Bill Greiman works well. See [this tutorial from Adafruit Industries](#) for instructions (thanks Limor!).

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general i/o. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 ethernet jack.

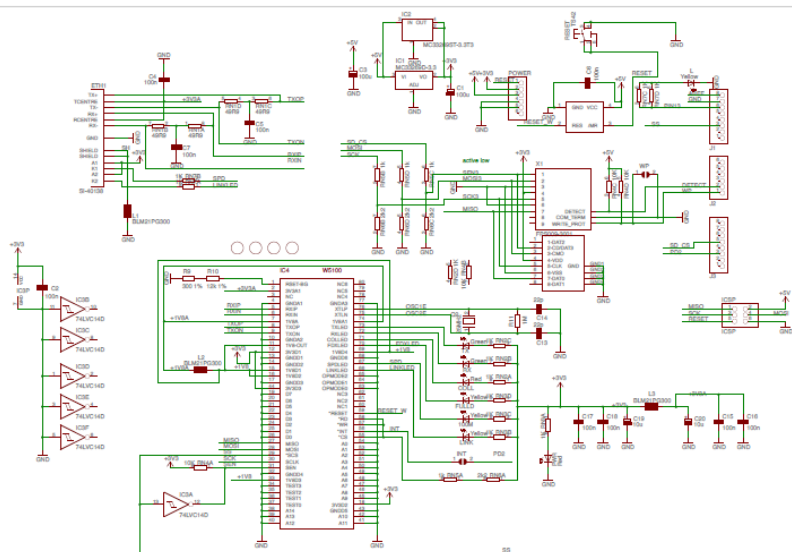
The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.

See also: [getting started with the ethernet shield](#) and [Ethernet library reference](#)



Arduino ETHERNET - shield V5

Copyright (c) 2010 Arduino
Released under the Creative Commons Attribution-Share Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0/>

B.3 Datasheet RTC DS1307

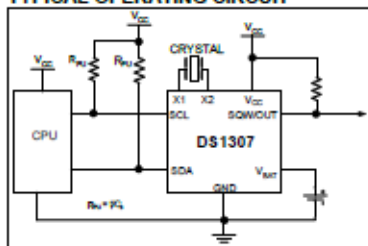


DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

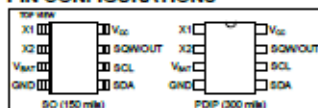
TYPICAL OPERATING CIRCUIT



BENEFITS AND FEATURES

- Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100
 - 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
 - Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
 - I²C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
 - Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
 - Automatic Power-Fail Detect and Switch Circuitry
- 8-Pin DIP and 8-Pin SO Minimizes Required Space
- Optional Industrial Temperature Range: -40°C to +85°C Supports Operation in a Wide Range of Applications
- Underwriters Laboratories® (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

*Denotes a lead-free/Pb-free compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device. Underwriters Laboratories, Inc. is a registered certification mark of Underwriters Laboratories, Inc.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial0°C to +70°C
Industrial-40°C to +85°C
Storage Temperature Range-55°C to +125°C
Soldering Temperature (DIP, leads)+260°C for 10 seconds
Soldering Temperature (surface mount)Refer to the JPC/JEDEC J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

($T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Logic 1 Input	V_{IH}		2.2		$V_{CC} + 0.3$	V
Logic 0 Input	V_{IL}		-0.3		+0.8	V
V_{BAT} Battery Voltage	V_{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 4.5\text{V}$ to 5.5V ; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I_{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I_{LO}		-1		1	μA
Logic 0 Output ($I_{OL} = 5\text{mA}$)	V_{OL}				0.4	V
Active Supply Current ($f_{SCL} = 100\text{kHz}$)	I_{CCA}				1.5	mA
Standby Current	I_{COS}	(Note 3)			200	μA
V_{BAT} Leakage Current	$I_{BAT,LEAK}$			5	50	nA
Power-Fail Voltage ($V_{BAT} = 3.0\text{V}$)	V_{PF}		$1.216 \times V_{BAT}$	$1.25 \times V_{BAT}$	$1.284 \times V_{BAT}$	V

DC ELECTRICAL CHARACTERISTICS

($V_{CC} = 0\text{V}$, $V_{BAT} = 3.0\text{V}$; $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$, $T_A = -40^\circ\text{C}$ to $+85^\circ\text{C}$.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V_{BAT} Current (OSC ON; SQW/OUT OFF)	I_{BAT1}			300	500	nA
V_{BAT} Current (OSC ON; SQW/OUT ON (32kHz))	I_{BAT2}			450	600	nA
V_{BAT} Data-Retention Current (Oscillator Off)	I_{BATER}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t _{SUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HDSTA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SUSTA}		4.7			μs
Data Hold Time	t _{HDAT}		0			μs
Data Setup Time	t _{SUDAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _R				1000	ns
Fall Time of Both SDA and SCL Signals	t _F				300	ns
Setup Time for STOP Condition	t _{SUSTO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{IO}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** f_{SCL} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{OH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{HDAT} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

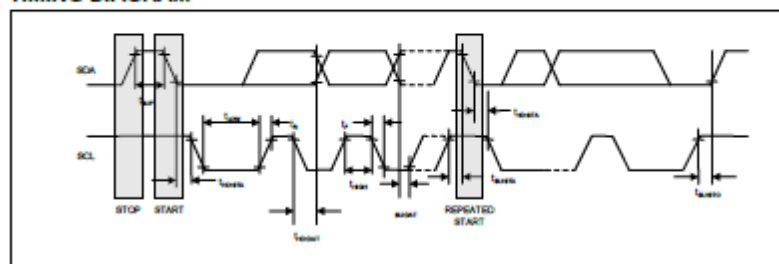
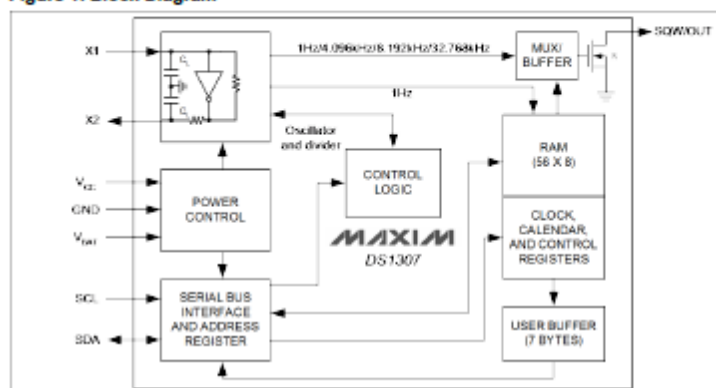
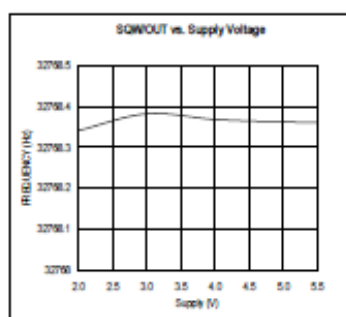
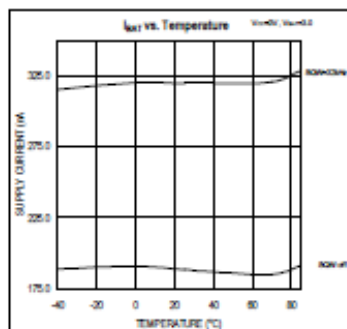
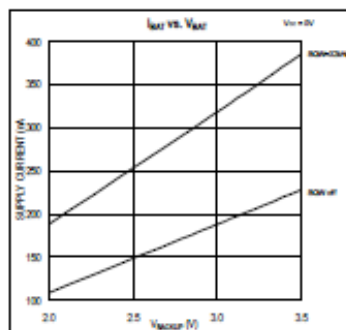
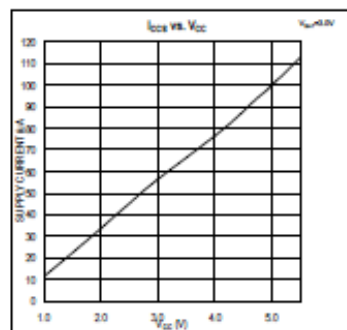


Figure 1. Block Diagram



TYPICAL OPERATING CHARACTERISTICS(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)

PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	Note: For more information on crystal selection and crystal layout considerations, refer to <i>Application Note 58: Crystal Considerations with Dallas Real-Time Clocks</i> .
3	V _{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V _{PF}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/ga/info/ul/ .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
6	SCL	Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWIE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . If not used, this pin can be left floating.
8	V _{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{PF} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} + 0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

B.4 Datasheet LCD 20x4

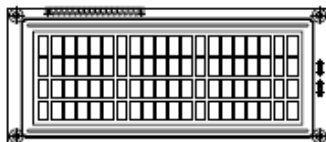


www.vishay.com

LCD-020N004L

Vishay

20 x 4 Character LCD



FEATURES

- Type: Character
- Display format: 20 x 4 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- +5 V power supply (also available for +3 V)
- LED can be driven by pin 1, pin 2, pin 15, pin 16 or A and K
- N.V. optional for +3 V power supply
- Material categorization: For definitions of compliance please see www.vishay.com/doc-7099012



MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	146.0 x 62.5	mm
Viewing Area	129.5 x 43.0	
Dot Size	0.92 x 1.10	
Dot Pitch	0.96 x 1.16	
Mounting Hole	139.0 x 55.5	
Character Size	4.84 x 9.22	

ABSOLUTE MAXIMUM RATINGS					
ITEM	SYMBOL	STANDARD VALUE			UNIT
Power Supply	V_{DD} to V_{SS}	MIN.	TYP.	MAX.	
Input Voltage	V_i	-0.5	-	7.0	V

Note

- $V_{SS} = 0$ V, $V_{DD} = \pm 0$ V

ELECTRICAL CHARACTERISTICS					
ITEM	SYMBOL	CONDITION	STANDARD VALUE		
			MIN.	TYP.	MAX.
Input Voltage	V_{DD}	$V_{DD} = \pm 5$ V	4.7	5.0	5.3
Supply Current	I_{DD}	$V_{DD} = +3$ V	2.7	3.0	5.3
		$V_{DD} = \pm 5$ V	-	8.0	10.0
Recommended LC Driving Voltage for Normal Temperature Version Module	V_{DD} to V_i	-20 °C	5.0	5.1	5.7
		0 °C	4.8	4.8	5.2
		25 °C	4.1	4.5	4.7
		50 °C	3.9	4.2	4.5
		70 °C	3.7	3.9	4.3
LED Forward Voltage	V_f	25 °C	-	4.2	4.6
LED Forward Current	I_f	25 °C	-	540	1080
EL Power Supply Current	I_{EL}	$V_{DD} = 110$ V _{AC} , 400 Hz	-	-	5.0

OPTIONS					
PROCESS COLOR					
TN	STN Gray	STN Yellow	STN Blue	FSTN B&W	STN Color
X	X	X	X	X	X

BACKLIGHT			
None	LED	EL	CCFL
X	X	X	X

For detailed information, please see the "Product Numbering System" document.

DISPLAY CHARACTER ADDRESS CODE																											
Display Position																											
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	20
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	60
DD RAM Address	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
DD RAM Address	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74

Revision: 09-10-12

1

Document Number: 37314

For technical questions, contact: display@vishay.com

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT www.vishay.com/doc-7099012

B.5 Datasheet ACS712

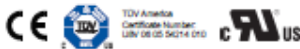


ACS712

*Fully Integrated, Hall Effect-Based Linear Current Sensor IC
with 2.1 kV_{RMS} Isolation and a Low-Resistance Current Conductor*

Features and Benefits

- Low-noise analog signal path
- Device bandwidth is set via the new FILTER pin
- 5 μ s output rise time in response to step input current
- 80 kHz bandwidth
- Total output error 1.5% at $T_A = 25^\circ\text{C}$
- Small footprint, low-profile SOIC8 package
- 1.2 m Ω internal conductor resistance
- 2.1 kV_{RMS} minimum isolation voltage from pins 1-4 to pins 5-8
- 5.0 V, single supply operation
- 66 to 185 mV/A output sensitivity
- Output voltage proportional to AC or DC currents
- Factory-trimmed for accuracy
- Extremely stable output offset voltage
- Nearly zero magnetic hysteresis
- Ratometric output from supply voltage



ISO 9001
TUV America
Certificate Number
L89 08 US 54214 D10



Package: 8 Lead SOIC (suffix LC)



Approximate Scale 1:1



Description

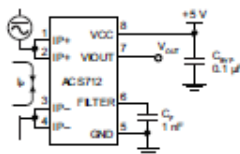
The Allegro™ ACS712 provides economical and precise solutions for AC or DC current sensing in industrial, commercial, and communications systems. The device package allows for easy implementation by the customer. Typical applications include motor control, load detection and management, switch-mode power supplies, and overcurrent fault protection. The device is not intended for automotive applications.

The device consists of a precise, low-offset, linear Hall circuit with a copper conduction path located near the surface of the die. Applied current flowing through this copper conduction path generates a magnetic field which the Hall IC converts into a proportional voltage. Device accuracy is optimized through the close proximity of the magnetic signal to the Hall transducer. A precise, proportional voltage is provided by the low-offset, chopper-stabilized BiCMOS Hall IC, which is programmed for accuracy after packaging.

The output of the device has a positive slope ($>V_{\text{OUT}/Q}$) when an increasing current flows through the primary copper conduction path (from pins 1 and 2, to pins 3 and 4), which is the path used for current sampling. The internal resistance of this conductive path is 1.2 m Ω typical, providing low power loss. The thickness of the copper conductor allows survival of

Continued on the next page...

Typical Application



Application 1. The ACS712 outputs an analog signal, V_{OUT} , that varies linearly with the uni- or bi-directional AC or DC primary sampled current, I_p , within the range specified. C_p is recommended for noise management, with values that depend on the application.

ACS712

Fully Integrated, Hall Effect-Based Linear Current Sensor IC with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor

Description (continued)

the device at up to 5x overcurrent conditions. The terminals of the conductive path are electrically isolated from the signal leads (pins 5 through 8). This allows the ACS712 to be used in applications requiring electrical isolation without the use of opto-isolators or other costly isolation techniques.

The ACS712 is provided in a small, surface mount SOIC8 package. The leadframe is plated with 100% matte tin, which is compatible with standard lead(Pb) free printed circuit board assembly processes. Internally, the device is Pb-free, except for flip-chip high-temperature Pb-based solder balls, currently exempt from RoHS. The device is fully calibrated prior to shipment from the factory.

Selection Guide

Part Number	Packing*	T _A (°C)	Optimized Range, I _P (A)	Sensitivity, Sens (Typ) (mV/A)
ACS712ELCTR-05B-T	Tape and reel, 3000 pieces/reel	-40 to 85	±5	185
ACS712ELCTR-20A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±20	100
ACS712ELCTR-30A-T	Tape and reel, 3000 pieces/reel	-40 to 85	±30	66

*Contact Allegro for additional packing options.

Absolute Maximum Ratings

Characteristic	Symbol	Notes	Rating	Units
Supply Voltage	V _{CC}		8	V
Reverse Supply Voltage	V _{ISO}		-0.1	V
Output Voltage	V _{OUT}		8	V
Reverse Output Voltage	V _{OUT}		-0.1	V
Output Current Source	I _{OUT(SOURCE)}		3	mA
Output Current Sink	I _{OUT(SINK)}		10	mA
Overcurrent Transient Tolerance	I _P	1 pulse, 100 ms	100	A
Nominal Operating Ambient Temperature	T _A	Range E	-40 to 85	°C
Maximum Junction Temperature	T _{J(max)}		165	°C
Storage Temperature	T _{stg}		-65 to 170	°C

Isolation Characteristics

Characteristic	Symbol	Notes	Rating	Unit
Dielectric Strength Test Voltage*	V _{ISO}	Agency type-tested for 60 seconds per UL standard 60950-1, 1st Edition	2100	VAC
Working Voltage for Basic Isolation	V _{WBI}	For basic (single) isolation per UL standard 60950-1, 1st Edition	354	VDC or V _{pk}
Working Voltage for Reinforced Isolation	V _{WRI}	For reinforced (double) isolation per UL standard 60950-1, 1st Edition	184	VDC or V _{pk}

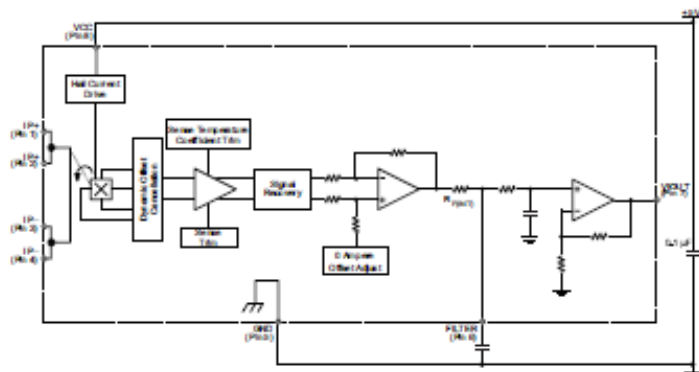
* Allegro does not conduct 60-second testing. It is done only during the UL certification process.

Parameter	Specification
Fire and Electric Shock	CAN/CSA-C22.2 No. 60950-1-03 UL 60950-1:2003 EN 60950-1:2001



Allegro MicroSystems, LLC
1155 Northwest Court
Worcester, Massachusetts 01615-0001 U.S.A.
1.508.853.3000, www.allegromicro.com

Functional Block Diagram



Pin-out Diagram



Terminal List Table

Number	Name	Description
1 and 2	IP+	Terminals for current being sampled, fused internally
3 and 4	IP-	Terminals for current being sampled, fused internally
5	GND	Signal ground terminal
6	FILTER	Terminal for external capacitor that sets bandwidth
7	VOUT	Analog output signal
8	VCC	Device power supply terminal

ACS712

*Fully Integrated, Hall Effect-Based Linear Current Sensor IC
with 2.1 kVRMS Isolation and a Low-Resistance Current Conductor*

COMMON OPERATING CHARACTERISTICS¹ over full range of T_A , $C_P = 1$ nF, and $V_{CC} = 5$ V, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
ELECTRICAL CHARACTERISTICS						
Supply Voltage	V_{CC}		4.5	5.0	5.5	V
Supply Current	I_{CC}	$V_{CC} = 5.0$ V, output open	–	10	13	mA
Output Capacitance Load	C_{LOAD}	VICOUT to GND	–	–	10	nF
Output Resistive Load	R_{LOAD}	VICOUT to GND	4.7	–	–	kΩ
Primary Conductor Resistance	$R_{PRIMARY}$	$T_A = 25^\circ\text{C}$	–	1.2	–	mΩ
Rise Time	t_r	$I_P = I_P(\text{max})$, $T_A = 25^\circ\text{C}$, $C_{OUT} = \text{open}$	–	3.5	–	μs
Frequency Bandwidth	f	–3 dB, $T_A = 25^\circ\text{C}$; I_P is 10 A peak-to-peak	–	80	–	kHz
Nonlinearity	E_{NL}	Over full range of I_P	–	1.5	–	%
Symmetry	E_{SYW}	Over full range of I_P	98	100	102	%
Zero Current Output Voltage	$V_{OUT(0)}$	Bidirectional; $I_P = 0$ A, $T_A = 25^\circ\text{C}$	–	$V_{CC} \times 0.5$	–	V
Power-On Time	t_{PO}	Output reaches 90% of steady-state level, $T_J = 25^\circ\text{C}$, 20 A present on leadframe	–	35	–	μs
Magnetic Coupling ²			–	12	–	G/A
Internal Filter Resistance ³	$R_{C(FILT)}$		–	1.7	–	kΩ

¹Device may be operated at higher primary current levels, I_P , and ambient, T_A , and internal leadframe temperatures, T_A , provided that the Maximum Junction Temperature, $T_J(\text{max})$, is not exceeded.

² $I_Q = 0.1$ mT.

³ $R_{C(FILT)}$ forms an RC circuit via the FILTER pin.

COMMON THERMAL CHARACTERISTICS¹

			Min.	Typ.	Max.	Units
Operating Internal Leadframe Temperature	T_A	E range	–40	–	85	°C
					Value	Units
Junction-to-Lead Thermal Resistance ²	$R_{JA(L)}$	Mounted on the Allegro A5EK 712 evaluation board			5	°C/W
Junction-to-Ambient Thermal Resistance	$R_{JA(A)}$	Mounted on the Allegro 85-0322 evaluation board, includes the power consumed by the board			23	°C/W

¹Additional thermal information is available on the Allegro website.

²The Allegro evaluation board has 1500 mm² of 2 oz. copper on each side, connected to pins 1 and 2, and to pins 3 and 4, with thermal vias connecting the layers. Performance values include the power consumed by the PCB. Further details on the board are available from the Frequently Asked Questions document on our website. Further information about board design and thermal performance also can be found in the Applications Information section of this datasheet.



Allegro Microsystems, LLC
115 Northwest Court
Worcester, Massachusetts 01615-0001 U.S.A.
1.508.653.5000; www.allegromicro.com

x05B PERFORMANCE CHARACTERISTICS¹ $T_A = -40^{\circ}\text{C}$ to 85°C , $C_P = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-5	-	5	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	180	185	190	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 185 mV/A programmed Sensitivity, $C_P = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	21	-	mV
Zero Current Output Slope	$\Delta V_{OUT(0)}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.28	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.054	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.008	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 5\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_P , with $I_P = 5\text{ A}$. Output filtered.

x20A PERFORMANCE CHARACTERISTICS¹ $T_A = -40^{\circ}\text{C}$ to 85°C , $C_P = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-20	-	20	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	96	100	104	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 100 mV/A programmed Sensitivity, $C_P = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	11	-	mV
Zero Current Output Slope	$\Delta V_{OUT(0)}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.34	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.07	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.017	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.004	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 20\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_P , with $I_P = 20\text{ A}$. Output filtered.

x30A PERFORMANCE CHARACTERISTICS¹ $T_A = -40^{\circ}\text{C}$ to 85°C , $C_P = 1\text{ nF}$, and $V_{CC} = 5\text{ V}$, unless otherwise specified

Characteristic	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Optimized Accuracy Range	I_P		-30	-	30	A
Sensitivity	Sens	Over full range of I_P , $T_A = 25^{\circ}\text{C}$	63	66	69	mV/A
Noise	$V_{NOISE(PP)}$	Peak-to-peak, $T_A = 25^{\circ}\text{C}$, 66 mV/A programmed Sensitivity, $C_P = 47\text{ nF}$, $C_{OUT} = \text{open}$, 2 kHz bandwidth	-	7	-	mV
Zero Current Output Slope	$\Delta V_{OUT(0)}$	$T_A = -40^{\circ}\text{C}$ to 25°C	-	-0.35	-	mV/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.08	-	mV/°C
Sensitivity Slope	ΔSens	$T_A = -40^{\circ}\text{C}$ to 25°C	-	0.007	-	mV/A/°C
		$T_A = 25^{\circ}\text{C}$ to 150°C	-	-0.002	-	mV/A/°C
Total Output Error ²	E_{TOT}	$I_P = \pm 30\text{ A}$, $T_A = 25^{\circ}\text{C}$	-	± 1.5	-	%

¹Device may be operated at higher primary current levels, I_P , and ambient temperatures, T_A , provided that the Maximum Junction Temperature, $T_{J(max)}$, is not exceeded.

²Percentage of I_P , with $I_P = 30\text{ A}$. Output filtered.



Allegro Microsystems, LLC
1155 Northwood Court
Worcester, Massachusetts 01615-0038 U.S.A.
1.508.853.5000; www.allegromicro.com

RIWAYAT HIDUP PENULIS



Nama : Rizqi Widya S.
TTL : Surabaya,
13 September 1996
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat : Jl. Taman Borobudur
No 18, Surabaya
Telp/HP : 081333572550
E-mail : *rizqiwidya76@gmail.com*

RIWAYAT PENDIDIKAN:

1. 2003-2009 : SD Negeri Ketabang 1 Surabaya
2. 2009-2012 : SMP Negeri 29 Surabaya
3. 2012-2015 : SMA Negeri 7 Surabaya
4. 2015-sekarang : Program Studi Elektro Industri,
Departemen Teknik Elektro Otomasi,
Fakultas Vokasi, Institut Teknologi
Sepuluh Nopember, Surabaya

PENGALAMAN KERJA

1. Kerja Praktek di PT PLN Area Surabaya Selatan (Persero)
Surabaya
2. Kerja Praktek di AirNav Juanda Surabaya (Persero)

PENGALAMAN ORGANISASI

1. Kewirausahaan HIMAD3TEKTRO 2016-2017
2. Kewirausahaan HIMAD3TEKTRO 2017-2018

-----Halaman ini sengaja dikosongkan-----

UCAPAN TERIMA KASIH

Puji syukur senantiasa penulis panjatkan puji syukur kepada Allah SWT, atas anugerahNya penulis dapat menyelesaikan tugas akhir ini. Pada kesempatan ini penulis ingin mengucapkan terima kasih sebesar- besarnya kepada orang- orang yang selalu membantu penulis dari awal perencanaan pembuatan hingga sampai akhir ini. Penulis mengucapkan terima kasih kepada :

1. Allah SWT atas segala karunianya penulis masih tetap diberi kesehatan dan rizki yang barokah sampai terselesaikan tugas akhir ini
2. Kedua orang tua penulis yang selalu mendukung, dan memberi doa kepada penulis hingga saat ini
3. Bapak Ir. Josaphat Pramudijanto, M. Eng. atas ketulusan beliau dalam meluangkan waktu untuk penulis dalam memberi ilmu, moral, dan spiritual dalam mengerjakan tugas akhir ini dari awal sampai akhir
4. Teman-teman EI 19 sudah selalu menemani, menyemangati dan memberi masukan di sela-sela pengerjaan tugas akhir
5. Teman-teman angkatan HYDRA 2015 yang selalu menemani, menyemangati penulis hingga tugas akhir ini selesai.

Sekali lagi penulis ucapkan terima kasih kepada seluruh orang yang telah membantu dan mendukung penulis. Penulis berharap semoga Allah SWT membalas kebaikan kalian semua dengan rahmat dan anugerahNya, Amin.